IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

FINJAN SOFTWARE, LTD., an Israel corporation,)
)
Plaintiff,) C. A. No. 06-369-GMS
)
v.)
)
SECURE COMPUTING CORPORATION, a Delaware corporation, CYBERGUARD, CORPORATION, a Delaware corporation, WEBWASHER AG, a German corporation and DOES 1 THROUGH 100,)
)
Defendants.)

APPENDIX OF EXHIBITS TO DECLARATION OF KRISTOPHER KASTENS IN SUPPORT OF PLAINTIFF FINJAN SOFTWARE, LTD.'S POST-TRIAL MOTION FOR INVALIDITY OF U.S. PATENT NO. 7,185,361 PURSUANT TO FED. R. CIV. P. 50(b)

VOLUME 1 – EXHIBITS 1-3 (PART 1)

Philip A. Rovner (#3215)
POTTER ANDERSON & CORROON LLP
Hercules Plaza
P. O. Box 951
Wilmington, DE 19899
(302) 984-6000
provner@potteranderson.com

Attorneys for Plaintiff
Finjan Software, Ltd.

OF COUNSEL:

Paul J. Andre
Lisa Kobialka
King & Spalding LLP
1000 Bridge Parkway
Redwood City, CA 94065
(650) 590-0700

Dated: April 25, 2008

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

CERTIFICATE OF SERVICE

I, Philip A. Rovner, hereby certify that on April 25, 2008, the within document was filed with the Clerk of the Court using CM/ECF which will send notification of such filing(s) to the following; that the document was served on the following counsel as indicated; and that the document is available for viewing and downloading from CM/ECF.

BY HAND DELIVERY AND E-MAIL

Frederick L. Cottrell, III, Esq.
Kelly E. Farnan, Esq.
Richards, Layton & Finger, P.A.
One Rodney Square
920 N. King Street
Wilmington, DE 19801
cottrell@rlf.com; farnan@rlf.com

I hereby certify that on April 25, 2008 I have sent by E-mail the foregoing document to the following non-registered participants:

Jake M. Holdreith, Esq.
Christopher A. Seidl, Esq.
Robins, Kaplan, Miller & Ciresi L.L.P.
2800 LaSalle Plaza
800 LaSalle Avenue
Minneapolis, MN 55402
jmholdreith@rkmc.com ; caseidl@rkmc.com

/s/ Philip A. Rovner
Philip A. Rovner (#3215)
Potter Anderson & Corroon LLP
Hercules Plaza
P.O. Box 951
Wilmington, Delaware 19899
(302) 984-6000
E-mail: provner@potteranderson.com

EXHIBIT 1

US007185361B1

(12) United States Patent
Ashoff et al.

(10) Patent No.: **US 7,185,361 B1**
(45) Date of Patent: **Feb. 27, 2007**

(54) SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR AUTHENTICATING USERS USING A LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP) DIRECTORY SERVER

(75) Inventors: Thomas D. Ashoff, Mt. Airy, MD (US); Steve O. Chew, Pittsburgh, PA (US); Jeffrey J. Graham, Olney, MD (US); Andrew J. Mullican, Columbia, MD (US)

(73) Assignee: Secure Computing Corporation, St. Paul, MN (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: 09/495,157

(22) Filed: Jan. 31, 2000

(51) Int. Cl.
H02H 3/05 (2006.01)

(52) U.S. Cl. 726/4; 713/151; 713/154; 707/1; 726/2; 726/8; 726/11; 726/12; 726/13; 726/14

(58) Field of Classification Search 713/201, 713/151–154; 707/1; 726/4, 8, 11–14
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

5,657,390	A *	8/1997	Elgamal et al.	713/151
5,898,830	A *	4/1999	Wesinger, Jr. et al.	713/201
6,047,322	A *	4/2000	Vaid et al.	709/224
6,131,120	A *	10/2000	Reid	709/225
6,182,142	B1 *	1/2001	Win et al.	709/229
6,212,558	B1 *	4/2001	Antur et al.	709/221
6,233,688	B1 *	5/2001	Montenegro	713/201
6,324,648	B1 *	11/2001	Grantges, Jr.	713/201
2003/0126468	A1 *	7/2003	Markham	713/201

OTHER PUBLICATIONS

Microsoft Corporation, Microsoft Computer Dictionary, Microsoft Press, Third edition, p. 197.*
Definition of application gateway, Webopedia computer dictionary, http://www.webopedia.com/TERM/A/application_gateway.html.*
Definition of firewall, Webopedia computer dictionary, http://www.webopedia.com/TERM/firewall.html.*
Netegrity, SiteMinder 3.5 Architecture.
How to Securely Manage and Control User Access to E-Commerce Web Sites, Netegrity White Paper, Jul. 1999.
Check Point Account Management Client, Version 1.0, Sep. 1998.
FireWall-1 Architecture and Administration; Chapter 4, pp. 135-154, Sep. 1998.
Howes et al., *The LDAP Application Program Interface*, University of Michigan, Aug. 1995.

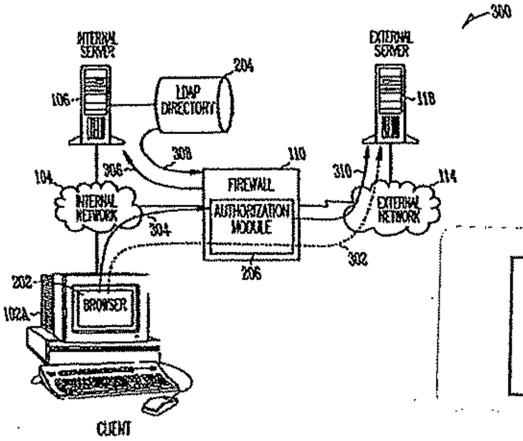
* cited by examiner

Primary Examiner—Taghi T. Arani
(74) *Attorney, Agent, or Firm*—Schwegman, Lundberg, Woessner & Kluth, P.A.

(57) ABSTRACT

A system, method and computer program product for providing authentication to a firewall using a lightweight directory access protocol (LDAP) directory server is disclosed. The firewall can be configured through a graphical user interface to implement an authentication scheme. The authentication scheme is based upon a determination of whether at least part of one or more LDAP entries satisfy an authorization filter.

15 Claims, 5 Drawing Sheets



Joint Trial Exhibit
JTX-5
Case No. 06-369 GMS

PENGAD 800-631-6989
EXHIBIT
12
GERMANY

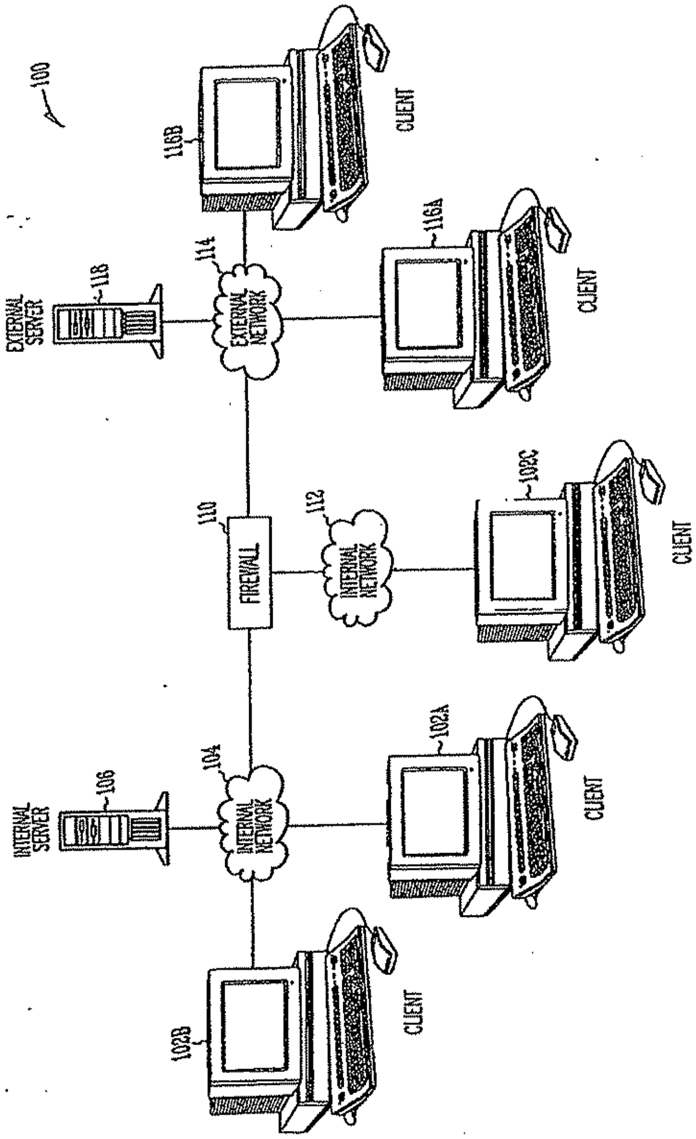


Fig.1 (Prior Art)

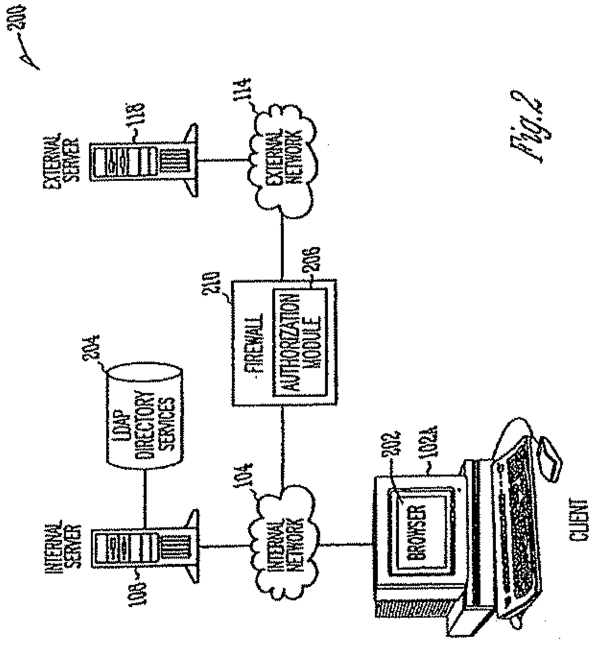

200
EXTERNAL SERVER
118
114
EXTERNAL NETWORK
210
FIREWALL
206
AUTHORIZATION MODULE
204
LDAP DIRECTORY SERVICES
INTERNAL SERVER
108
104
INTERNAL NETWORK
202
102A
BROWSER
CLIENT

Fig.2

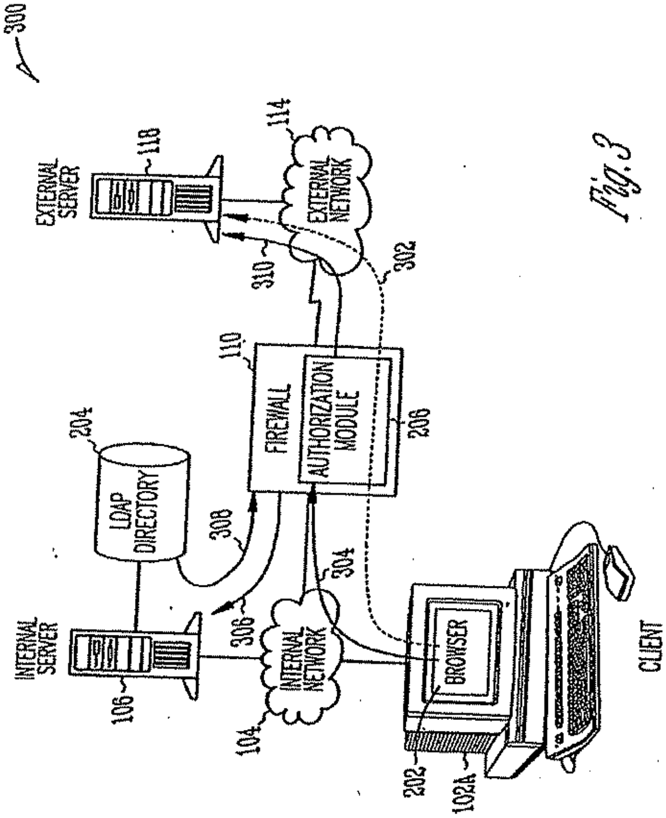

300
EXTERNAL SERVER
118
114
EXTERNAL NETWORK
310
302
110
FIREWALL
AUTHORIZATION MODULE
206
204
LDAP DIRECTORY
308
304
306
INTERNAL SERVER
INTERNAL NETWORK
106
104
BROWSER
202
102A
CLIENT

Fig. 3

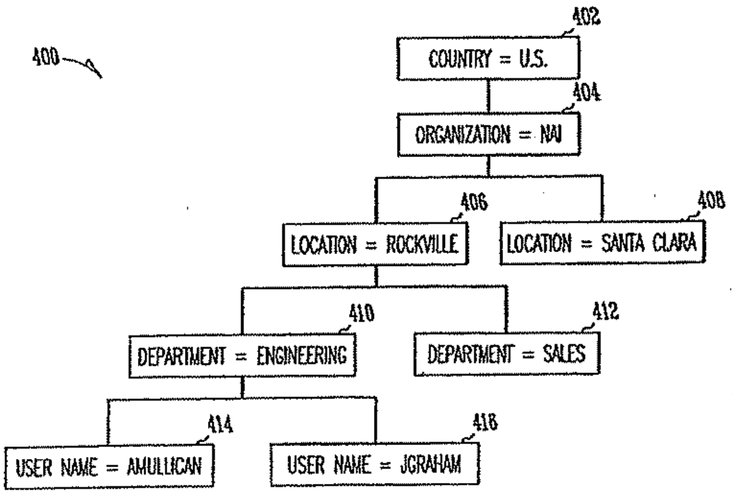

400
402
COUNTRY = U.S.
404
ORGANIZATION = NAI
406
408
LOCATION = ROCKVILLE
LOCATION = SANTA CLARA
410
412
DEPARTMENT = ENGINEERING
DEPARTMENT = SALES
414
416
USER NAME = AMULLICAN
USER NAME = JGRAHAM

Fig.4

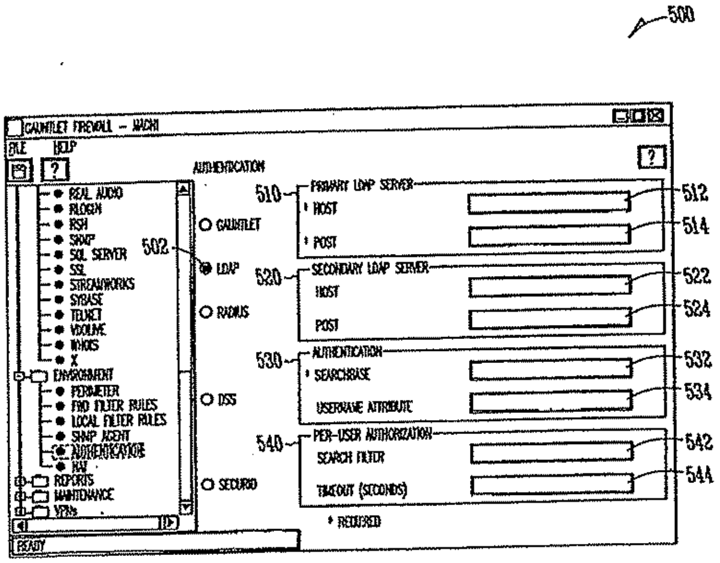

500
GAUNTLET FIREWALL – MACHI
FILE
HELP
?
?
AUTHENTICATION
REAL AUDIO
RLOGIN
RSH
SNMP
SQL SERVER
SSL
STREAMWORKS
SYBASE
TELNET
VDOLIVE
WHOIS
X
ENVIRONMENT
PERIMETER
FWD FILTER RULES
LOCAL FILTER RULES
SNMP AGENT
AUTHENTICATION
NAT
REPORTS
MAINTENANCE
VPNs
READY
502
○ GAUNTLET
● LDAP
○ RADIUS
○ DSS
○ SECURID
510
PRIMARY LDAP SERVER
* HOST
* POST
512
514
520
SECONDARY LDAP SERVER
HOST
POST
522
524
530
AUTHENTICATION
* SEARCHBASE
USERNAME ATTRIBUTE
532
534
540
PER-USER AUTHORIZATION
SEARCH FILTER
TIMEOUT (SECONDS)
542
544
* REQUIRED

Fig.5

SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR AUTHENTICATING USERS USING A LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP) DIRECTORY SERVER

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to user authentication mechanisms and more particularly to user authentication mechanisms for firewalls.

2. Related Art

Control over access to information technology (IT) resources is a common need today. A firewall can be used to protect IT resources behind the firewall. Network firewalls can enforce a site's security policy by controlling the flow of traffic between two or more networks. For example, a company might encourage file transfers to the company's network that assist employees, but might discourage file transfers of potentially sensitive company confidential information from the company network to external destinations. Firewalls often are placed between a corporate network and an external network such as, e.g., the Internet, or a partnering company's network. Firewalls can also be used to segment parts of a corporate network. A firewall system can provide both a perimeter defense to, e.g., an internal network, and a control point for monitoring access to and from specific networks such as, e.g., an external network.

Firewalls can control access at a network level, an application level, or both. At the network level, a firewall can restrict packet flow based on protocol attributes. For example, the packet's source address, destination address, originating transmission control protocol/user datagram protocol (TCP/UDP) port, destination port, and protocol type can be used for the control decisions. At an application level, a firewall can participate in communications between the source and destination applications with the firewall's control decisions being based on details of the conversation and other available information such as, e.g., previous connectivity or user identification. Thus, a firewall can authenticate users to control access to and from IT resources behind and before the firewall.

Firewalls can be packaged as system software, combined hardware and software, and, more recently, dedicated hardware appliances (e.g., embedded in routers, or easy-to-configure integrated hardware and software packages that can run on dedicated platforms). An example of an application-based firewall is the Gauntlet™ firewall available from Network Associates, Inc.

Firewalls can defend against attacks ranging from, e.g., unauthorized access, IP address "spoofing" (i.e., a technique by which hackers disguise traffic as coming from a trusted address to gain access to a protected network or resource), buffer overrun attacks, session hijacking, viruses and rogue applets, and rerouting of traffic. However, inherent limitations exist in certain services and protocols that conventional firewalls cannot remedy.

Conventionally, when software application programs sought to restrict what a user could do with the programs, the programs required identification of the user. For example, if a user desires access to sensitive corporate financial data in an accounting program, access to the data can be restricted by means of authentication mechanisms such as, e.g., a password. The application program therefore requires a list of users and identification information for the user for use in authenticating the user.

Early software application programs often included their own integrated authentication mechanisms. Users often use a variety of software application programs, each possibly having its own authentication mechanism. Users find it cumbersome to remember different passwords associated with each of the multiple software application programs.

IT resources used by companies today can include access to multiple software application programs and Internet based applications. For example, employees at a given company can use e-mail and groupware applications, and other office automation programs including, e.g., to spreadsheets, wordprocessors and presentation programs. As every application program conventionally has its own authentication mechanism, a separate database is initialized and updated for each application.

Authentication mechanisms can use a query to a database known as a directory that can store information about users. A directory is similar to a database in that one can store information in a directory and later retrieve the information from it. However, a directory is specialized in that a directory is typically designed for reading more than writing. A directory offers a static view of the information and allows simple updates without transactions. Thus, while a database is typically written to and read from frequently, a directory by comparison is primarily read from and is infrequently updated.

A directory service includes all the functions of a directory and adds a network protocol that can be used to access the directory. Standardization is desirable in implementing a directory service.

An early standard for directory service was the directory access protocol (DAP), which originated in the European standards organization. DAP although specifying a vast, feature-rich protocol for storing and encoding directory information, was unwieldy in size.

Today, a new protocol, lightweight directory access protocol (LDAP), is gaining wide acceptance in business. The LDAP standard defines an information model for a directory, a namespace for defining how directory information is referenced and organized, and a network protocol for accessing information in the directory. LDAP can also include an application programming interface (API). The LDAP protocol mandates how client and server computers can communicate with a LDAP directory. However, LDAP does not mandate how data should be stored. More and more companies today use an LDAP directory server to store a database of employees. The LDAP directory generally can store an employee name, phone number, address and other information about the employee, and a password for modifying the employee's information.

Firewalls also maintain a database of users and are operative to prompt users for an identifying user identifier and password. These conventional firewalls require that employee names and passwords be entered into a firewall authentication database. Maintenance of the firewall authentication database is especially burdensome where there are a large number of employees that are frequently leaving or joining a company or when a company has a large number of firewalls. Accordingly, what is needed is a mechanism for reducing this administrative burden. More specifically, what is needed is a mechanism for leveraging an existing LDAP directory server as part of a firewall's authentication process. In this manner, an existing LDAP directory server can be used as a central directory that stores the data used by all applications.

SUMMARY OF THE INVENTION

A system, method and computer program product for enabling the authentication of users to a firewall using a lightweight directory access protocol (LDAP) directory server is provided by the present invention. The firewall can be configured through a graphical user interface to implement an authentication scheme. The authentication scheme is based upon a determination of whether information contained in one or more LDAP entries satisfy an authorization filter. It is a feature of the present invention that the authentication scheme can be configured independently of specifically stated field requirements or schema of the firewall. In accordance with the present invention, the authentication scheme can be flexibly specified to interact with a LDAP directory that has been uniquely developed for a company's internal needs. The company's investment in its existing administrative infrastructure can therefore be leveraged to a greater degree.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other features and advantages of the invention will be apparent from the following, more particular description of a preferred embodiment of the invention, as illustrated in the accompanying drawings.

FIG. 1 illustrates a communications network including a firewall.

FIG. 2 illustrates a communications network including a lightweight directory access protocol (LDAP) directory server and an authorization module within a firewall.

FIG. 3 illustrates the authentication of a client user through a firewall.

FIG. 4 illustrates an example embodiment of an LDAP directory tree.

FIG. 5 illustrates an embodiment of a graphical user interface for configuring the LDAP authentication feature.

DETAILED DESCRIPTION OF THE INVENTION

A preferred embodiment of the invention is discussed in detail below. While specific implementations are discussed, it should be understood that this is done for illustration purposes only. A person skilled in the relevant art will recognize that other components and configurations may be used without parting from the spirit and scope of the invention.

FIG. 1 illustrates an example embodiment of a communications network 100 including client computers 102*a* and 102*b* coupled via an internal network 104 to an internal server computer 106 and a firewall 110. Communications network 100 also includes a client computer 102*c* coupled via an internal network 112 to firewall 110. Finally, communications network 100 includes client computers 116*a* and 116*b* coupled via an external network 114 to an external server computer 118 and firewall 110. External network 114 can represent, e.g., the Global Internet, or a partnering company's network.

Network firewall 110 can enforce a business' security policy by controlling the flow of traffic between two or more networks such as, e.g., internal networks 104 and 112 and external network 114. In general, firewall 110 serves to isolate internal networks 104 and 112 from one another and also from external network 114.

As illustrated in FIG. 1, firewall 110 can be used to segment parts of a corporate network. For example, firewall 110 can be used to control information flow between a corporation's internal networks 104, 112. Firewall 110 can also provide a perimeter defense between an internal network 104, 112 and an external network 114.

FIG. 2 illustrates an example embodiment of a communications network 200 that includes client computer 102*a* coupled via internal network 104 to internal server 106 and to firewall 210. Firewall 210 is also coupled via external network 114 to external server 118.

As shown, client computer 102*a* includes a browser 202. Browser 202 can in one embodiment be an Internet browser that provides a graphical user interface to network resources. Browser 202 is generally operative to parse and make requests to network resources such as, e.g., external server 118, and present the results of the request to a client user viewing client computer 102*a*.

Internal server 106 is shown including a lightweight directory access protocol (LDAP) directory 204, which can be configured to store employee information. For example, a human resources database could be stored as an LDAP directory having a directory structure such as that illustrated in FIG. 4. As illustrated, LDAP directory tree 400 includes country 402 set in this example to US, organization 404 set to NAI, location 406 set to Rockville and location 408 set to Santa Clara, department 410 set to engineering and department 412 set to sales, and username 414 set to amullican and username 416 set to jgraham.

External server 118 can include an Internet server application. In one embodiment, the Internet server application supports file transfer protocol (FTP) communication. As would be apparent to those skilled in the relevant art, other types of server applications can be included on external server 118 including, e.g., databases, and electronic mail.

Firewall 210 is shown including an authorization module 206. Authorization module 206 is used to authenticate a client user (e.g., client computer 102*a*) to determine if the client user's communication is authorized to pass through firewall 210. Conventional firewalls 110 included their own database having a list of users and passwords, to enable authentication through firewall 110.

In accordance with the present invention, firewall 210 does not authenticate users using its own database. Rather, firewall 210 authenticates users using information contained within LDAP directory 204. As will be described in greater detail below, firewall 210 can authenticate users through an authentication scheme that can be based upon the unique composition of an organization's LDAP directory 204.

It is a feature of the present invention that the authentication scheme of the present invention can operate independently of specifically stated field requirements or schema of the firewall 210. In other words, an organization's LDAP directory 204 need not be modified to conform to a schema imposed by the firewall 210. Moreover, resistance to such a modification will not result in the maintenance of multiple directories.

In accordance with the present invention, the authentication scheme can be flexibly specified to interact with an existing LDAP directory that has been uniquely developed for a organization's internal needs. This framework enables a firewall administrator to seamlessly integrate a firewall product into an existing administrative infrastructure. The organization's investment in the existing administrative infrastructure can therefore be leveraged to a greater degree.

FIG. 3 illustrates the authentication process that is implemented by firewall 210. In the illustrated example, firewall 210 authenticates a client user at client computer 102*a* running a browser 202 that is attempting to access an application or resource on external server 118. This access path is illustrated by path 302.

This authentication process begins when client computer 102*a* initiates a network resource request 304 from browser 202. The network resource request 304 is intercepted by firewall 210. Authorization module 206 within firewall 210 challenges the client user to identify himself or herself. A challenge could in one embodiment include a request for entry of a username and password. Upon receipt of the identification information, authorization module 206 searches an authentication database (not shown) to identify an authentication method (e.g., LDAP authentication). If no entry in the authentication database is found for the client user, then a default authentication method can be used. In the LDAP authentication process, authorization module 206 binds to LDAP directory 204 and uses the userPassword attribute for authentication.

After authorization module 206 authenticates the client user, authorization module 206 then determines whether the client user is authorized to have his access request fulfilled. The LDAP authorization process is illustrated as communications 306 and 308. Communications 306 and 308 are facilitated using the LDAP protocol and may utilize the secure sockets layer.

If per-user authorization is configured, authorization module 206 determines whether one or more attributes of the client user's LDAP entry satisfies an authorization filter. If the one or more attributes of the client user's LDAP entry does not satisfy the authorization filter, then authorization module 206 determines that the authorization fails. If the authorization filter is satisfied, then the client user's network resource request is allowed through firewall 210. This allowed connection is illustrated in FIG. 3 as path 310.

To support per-user authorization, an administrator configures an authorization filter to use when authenticating users. One or more attributes in the client user's LDAP directory entry and associated values can be selected for the authorization filter. Once configured, authorization module 206 can verify that the LDAP entry used in the bind call satisfies the authorization filter before allowing the user access to/through the firewall.

FIG. 5 illustrates an example embodiment of a graphical user interface (GUI) 500 of a firewall systems administrator application screen. As shown by a selected radio button, LDAP authentication 502 has been selected. GUI 500 includes a primary LDAP server settings area 510, a secondary LDAP server settings area 520, an authentication settings area 530, and a per-user authorization settings area 540.

The primary LDAP server settings area 510 includes a host field 512 and a port field 514. The host field 512 can be used to enter an IP address or host name of a primary LDAP server. The port field 514 can be used to enter the port to be used on the primary LDAP server.

The secondary LDAP server settings area 520 also includes a host field 522 and a port field 524. The host field 522 can be used to enter an IP address or host name of a secondary LDAP server. The port field 524 is used to enter the port to be used on the secondary LDAP server. Fields 522, 524 can be left blank if no secondary LDAP server is being used.

The authentication settings area 530, can include searchbase field 532 and a username attribute field 534. The searchbase field 532 can be used to indicate the top of the directory tree 400 such as, e.g., country 402, organization 404, location 406, and department 410, so that a lookup can be within that portion of the directory tree. For example, a

set of attribute pairs such as, e.g., o=NAI, c=US to append to all requests to the LDAP server can be entered. The username attribute field 534 can include a default username attribute such as, e.g., uid. The username attribute field 534 can be used in performing per-user authorization.

The per-user authorization settings area 540 includes a search filter field 542 and a timeout field 544. The timeout field 544 can include a default value such as, e.g., 60 seconds. For example, timeout field 544 can be used to limit the amount of time the authorization filter query can take. If the time is exceeded, the authorization fails.

The search filter field 542 is used by firewall 210 in identifying the appropriate fields that are the subject of the LDAP directory authentication query. Upon receipt of a response from the LDAP directory 204, firewall 210 can then determine whether the client user is authorized to authenticate through the firewall 210.

In general, the authorization filter can contain any LDAP-valid combination of attributes and values, including object classes. At its simplest, the authorization filter specifies a single attribute and value pair. For example, the search filter field 542 can be used to enter a search filter expression such as "objectclass=gauntletUser."

Consider another example where LDAP directory 204 is configured by the company to include a field that would provide an access code level for each user. For example a "1" could correspond to only e-mail access, while a 5 could mean full access to all Internet services including world wide web browsing. In this environment, an authorization filter can be specified as "(&(objectclass=gUser)(status>=5))".

It should be noted that the authorization process need not be based on per-user authorization. In another embodiment, the authorization process can be based on a per-service authorization. In this embodiment, the per-service authorization can include an authorization for protocol services. Examples of protocol services include FTP, simple mail transport protocol (SMTP) e-mail, hypertext transport protocol (HTTP), etc. The per-service authorization can also be based on LDAP directory information. For example, authorization module 206 can use group memberships to determine whether a client user can use HTTP through firewall 210. To satisfy this authorization process, the authenticated user must be a member of the "web-users" group in the LDAP directory.

In one embodiment, the per-service authorization process uses the standard groupOfNames and groupOfUniqueNames object classes for authorization decisions. In general, a mechanism can be included that supports the specification of arbitrary group names for each service to be controlled. Control can then be based on a per-proxy basis or a per-policy basis.

Specification of per-service authorization criteria can also be implemented using the search filter field 542. In general, a different search (or authorization) filter can be provided for each service. For example, a search filter field can be included in GUI 500 to determine whether, e.g., a user is authorized to perform a file transfer, to send e-mail, or to access the world wide web. A search filter field can also be included in GUI 500 to determine whether, e.g., a user is a member of a particular group such as, e.g., engineering department 410, and if so, then particular services can be authorized based on being part of that group.

As noted, it is a feature of the present invention that firewall 210 can support arbitrary LDAP directory schema. Accordingly, firewall 210 does not require additional firewall-specific object classes or attributes in the directory.

Customers can populate the LDAP directories with whatever data they require. This authentication environment can be flexibly applied across multiple organizations each having their own sets of directory information. Indeed, the concepts of the present invention can be used to implement an authorization filter that relies on portions of information that are stored in distinct LDAP directories. This distributed authentication scheme enables an organization to implement segmented management of the user database.

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A system for authorizing client access to a network resource, comprising:

a server having at least one directory that can be accessed using a network protocol, said at least one directory being configured to store information concerning an entity's organization; and

a firewall that is configured to intercept network resource requests from a plurality of client users on an internal network, said firewall being operative to authorize a network resource request based upon a comparison of the contents of at least part of one or more entries in said at least one directory to an authorization filter, wherein said authorization filter is generated based on a directory schema that is predefined by said entity.

2. The system of claim 1, wherein said at least one directory is a lightweight directory access protocol directory.

3. The system of claim 1, wherein said authorization filter is specified using a graphical user interface.

4. The system of claim 1, wherein said authorization filter implements a per-user authentication scheme.

5. The system of claim 1, wherein said authorization filter implements a per-service authentication scheme.

6. The system of claim 1, wherein said firewall and said directory communicate using secure socket layer communication.

7. The system of claim 1, wherein said firewall is configured to query multiple directories.

8. An authentication method at a firewall, comprising the steps of:

(a) receiving a network resource request from a client user at an internal network;

(b) querying, using a network protocol, at least one directory that is configured to store information concerning an entity's organization, wherein said query is based upon an authorization filter that is generated based on a directory schema that is predefined by said entity;

(c) determining, based on the results of said query, whether the contents of at least part of one or more entries in said at least one directory satisfy said authorization filter; and

(d) permitting said network resource request through said firewall if said authorization filter is satisfied.

9. The method of claim 8, wherein step (b) comprises the step of querying said at least one directory using a lightweight directory access protocol.

10. The method of claim 8, further comprising the step of specifying an authorization filter using a graphical user interface.

11. The method of claim 10, wherein said specifying step comprises the step of specifying an authorization filter that implements a per-user authentication scheme.

12. The method of claim 10, wherein said specifying step comprises the step of specifying an authorization filter that implements a per-service authentication scheme.

13. The method of claim 8, wherein step (b) comprises the step of querying said directory using secure socket layer communication.

14. The method of claim 8, wherein step (b) comprises the step of querying multiple directories.

15. A computer program product for enabling a processor in a computer system to implement an authentication process, said computer program product comprising:

a computer usable medium having computer readable program code embodied in said medium for causing a program to execute on the computer system, said computer readable program code comprising:

first computer readable program code for enabling the computer system to receive a network resource request from a client user at an internal network;

second computer readable program code for enabling the computer system to query, using a network protocol, at least one directory that is configured to store information concerning an entity's organization, wherein said query is based upon an authorization filter that is generated based on a directory schema that is predefined by said entity;

third computer readable program code for enabling the computer system to determine, based on the results of said query, whether the contents of at least part of one or more entries in said at least one directory satisfy said authorization filter; and

fourth computer readable program code for enabling the computer system to permit said network resource request through a firewall if said authorization filter is satisfied.

* * * * *

EXHIBIT 2

1435

IN THE UNITED STATES DISTRICT COURT

IN AND FOR THE DISTRICT OF DELAWARE

- - -

FINJAN SOFTWARE LTD., : Civil Action
: No. 06-369(GMS)
Plaintiff, :
:
v. :
:
SECURE COMPUTING CORPORATION, :
CYBERGUARD CORPORATION, :
WEBWASHERE AG and DOES 1 :
THROUGH 100, :
:
Defendants. :

- - -

Wilmington, Delaware
Tuesday, March 11, 2008
8:50 a.m.
Day Seven of Trial

- - -

BEFORE: HONORABLE GREGORY M. SLEET, Chief Judge,
and a Jury

APPEARANCES:

PHILIP A. ROVNER, ESQ.
Potter Anderson & Corroon LLP
-and-
PAUL J. ANDRE, ESQ.,
LISA KOBIALKA, ESQ.,
JAMES HANNAH, ESQ.,
MEGHAN WARTON, ESQ.,
KRIS KASTENS, ESQ., and
HANNAH LEE, ESQ.
King & Spalding
(Silicon Valley, California)

Counsel for Plaintiff

1436

APPEARANCES (Continued):

FREDERICK R. COTTRELL, III, ESQ., and
KELLY E. FARNAN, ESQ.
Richards, Layton & Finger
-and-
RONALD J. SCHUTZ, ESQ.,
CHRISTOPHER A. SEIDL, ESQ.,
TREVOR J. FOSTER, ESQ., and
JAKE M. HOLDREITH, ESQ.
Robins, Kaplan, Miller & Ciresi, L.L.P.
(Minneapolis, MN)

Counsel for Defendants

1437

THE COURT: Good morning. Please be seated.

All right. Here are the rulings on the remaining jury instruction issues.

As to No. 16, I am going to side with Finjan's position on this. I was working from Secure's instruction. I am going to eliminate the entirety of the last paragraph except the last sentence, insert it as the last paragraph in the Finjan, in the proposed 16. You made a bit of a mess, Finjan, of the last sentence. You might want to proofread a little more carefully next time.

That means that 19.2 will not be given. That carries forward, the same line, same ruling will carry forward in the damages instruction as well.

As to 47, proposed Finjan 48, I am going to strike it. I am going to give Finjan's proposed 70. I don't know why you didn't number them both 47. I guess it was just to pluck my nerves. But we are going to give 48, Secure's instruction. I am rejecting Finjan's 47.

I think Secure's position is the better position, based on my reading of the law.

And there was a marking. I am going to overrule Secure's objection, go ahead and give the marking instruction. If I am wrong and you convince me I am wrong, I can correct that later on. But you are going to have to deal with that now. Right now, given the amount of time

1438

that I had to deal with the issues, that is the best rulings I can -- that is the way I see the rulings at this point.

All right. Are you ready for the jury?

With that, I expect that we can get the instructions in shape, give them to the other side, and get sufficient copies, and we will instruct them as soon as we are able.

The witness can resume the stand.

All parties' objections have been acknowledged by the Court and reserved. The Court has ruled as it has. Clearly, in my view, I am going to be very disappointed if either of you goes up to the Federal Circuit on any of these waiver issues. I don't think either party has waived anything. You may have waived the issue of marking. I know that is part of the marking. There is just no evidence. Insofar as preservation of issues for appeal, waivers, I just cannot imagine.

But maybe.

Again, just to recapitulate, with regard to closings, plaintiffs will have a total of an hour, the defense will have 45. If you want to rebut, you are going to have to reserve a portion of the hour.

MR. ANDRE: Thank you, Your Honor.

THE COURT: We are still waiting for one. Why don't we just relax for a few moments.

1459

A. I have taught two different graduate classes on computer security. One is relevant to all topics of security, photography, networks, and operating a net security code, operating and network security, that was a little more advanced. I have taught twice an undergraduate class on the broader topics of computer security. In this semester, I am teaching an operating class on computer security.

MS. KOBIALKA: Your Honor, at this time we would like to tender Dr. Jaeger as an expert in computer security.

MR. SCHUTZ: No objection, Your Honor.

THE COURT: He is accepted.

BY MS. KOBIALKA:

Q. Were you asked to provide an opinion as to whether Finjan infringed the '361 and '010 patents?

A. Yes, I was.

Q. Before you were asked to be an expert in this case, did you have an opinion on the topic?

A. No, I did not.

Q. Is that also the same with respect to your opinions on whether or not the same patents are valid?

A. That is correct. I had no opinion previously.

Q. Do you have an understanding that Secure Computing is alleging that the NG appliances infringe the '361 patent?

A. That is my understanding, yes.

1460

Q. When we refer to the NG appliances, do you understand that we are referring to the NG 1100, NG 5100, NG 6100 and NG 8100?

A. Yes, I do.

Q. Do you have an understanding of how those units function?

A. My understanding is they all basically have the same function.

Q. So I am just going to refer to them jointly as the NG appliances. Is that okay?

A. That is fine.

Q. We don't need to discuss each one separately?

A. That's right.

Q. Do you have an understanding that Finjan's Vital Security for Documents has been alleged to infringe a certain claim of the '010 patent?

A. That's right.

Q. Now, in forming your opinions with regard to all the things that you are going to discuss today, what did you review?

A. Quite a few things.

We have the patents themselves, prior art, the Finjan product documentation, the source code for the NG appliance, the patent prosecution history, the interaction between the inventors and the Patent Office in processing

1461

the patent, and some other things you might remind me of here.

Q. Did you review the materials that Dr. Wallach had reviewed with respect to some Vital Security for Documents fliers?

A. Oh, yeah.

So I reviewed the documents that were provided, it would be called the claim of infringement or something like this by Dr. Wallach, he provided a report, by Dr. Wallach.

Q. Did you review any deposition testimony?

A. I reviewed Dr. Wallach's testimony, yes.

Q. Are you referring to his trial testimony?

A. Yes.

Q. Did you review the deposition testimony of Mr. Chew?

A. Yes. I reviewed the deposition testimony of Mr. Chew, Ms. Greve, and then the two Finjan -- Mr. Ben-Itzhak and Mr. Frommer.

Q. In forming your opinion, did you use the interpretation of certain terms that the Court had provided regarding the patents?

A. Yes, I did.

Q. Now, are the technologies of the '361 and the '010 patent related to each other at all?

A. Not at all.

1462

Q. Let's turn then first to the '361 patent. I would like to pull up GP 361A.

Could you describe for the jury what type of technology is involved in the '361 patent?

A. Yes, I can.

So the '361 patent is about, it's about a system where you have a firewall and you have a directory server that the firewall uses to help it make its decisions about whether requests from the client computer, in this case individual packets, can be submitted through the firewall to the Internet.

So it's for outgoing requests from the client to the Internet. It is going to determine whether you can send something out from your network.

Q. You have a laser pointer, too, if it would help.

A. Okay.

So the first step -- I think we have a sequence here. The first step, the client is going to submit the request, and it's going to be broken down into a sequence of packets, and then sent along the network, and it's going to be intercepted by the firewall here.

Then we have the second step.

The firewall has to make a decision, can this client make this request?

The next, it's going to -- in order to make that

1491

Jaeger - direct

it may let you go through. It doesn't really talk about -- sorry.

It will succeed in authenticating and pass the authorization filter and let it go through.

Q. The NG appliances don't utilize an authentication method?

A. They do not implement an authentication method at a firewall consisting of those steps, no.

Q. Okay. So that, then, supports your opinion of noninfringement with regard to Claims 8 through 12 and 14. Correct?

A. Correct.

Q. Why don't we turn to the next slide for Claim 15. I believe this authentication process also appears in the beginning, where it says a computer program product for enabling a processor in a computer system to implement an authentication process.

A. That's correct.

Q. Is that the same type of thing we had just discussed?

A. Yes, it is.

Q. Is it also your opinion that there is no infringement of Claim 15 in connection with the authentication process?

A. That is correct.

Q. There is also some language, per service, that was used, I believe, in Claim 5. What does per service mean?

1492

Jaeger - direct

A. So a service is some, basically, program that is listening on the network. So there are services for logging into computers. There are services for sending e-mail. There are services for using a web. And so these define specific network facing is the term we will use, functionality that you can communicate with over the Internet.

So we have in the patent two types of authentication. One is per user, and one is per service. The idea -- the distinction they are making is that, in my opinion, is that you can authenticate as Alice or Bob. You can say, okay, I will be Bob this time. So you are Bob. And I want to gain access to the network. And so the firewall will run some scheme to authenticate that you are really Bob, and then based on determining that you are Bob, it will let you have whatever access that you want, that it will authorize.

The other thing is you may ask to authorize for a specific service, maybe for e-mail. So you will just ask to authorize for that particular service. So you are going to say I am going to authorize Bob to use the e-mail service. And that is all.

Q. Dr. Jaeger, do you have an understanding of what is meant by inducing infringement?

A. Yes.

1493

Jaeger - direct

Q. Do you have an opinion regarding whether Finjan has induced infringement of the asserted claims of the '361 patent as a result of its NG appliances?

A. My opinion is they have not induced infringement, the NG appliance has not induced infringement of the '361 claims.

Q. Has there been any evidence of any inducing infringement that you were able to read in Dr. Wallach's testimony?

A. I saw no specific case.

Q. Do you have any other bases for your opinion regarding no inducing of infringement by Finjan?

A. Yes, I do.

Q. Are those the same as you have discussed already this morning?

A. Yes.

Q. Why don't we turn to now the assertion of invalidity with regard to the '361 patent.

What was your determination regarding whether or not the asserted claims of the '361 patent were valid?

A. So my determination was that the claims of the '361 patent are invalid.

Q. Was it based on your theory of anticipation and obviousness?

A. Yes. It was based on both.

1494

Jaeger - direct

Q. What is your understanding of what we mean them by anticipation?

A. So by anticipating, my understanding is that this requires one reference to disclose or one system to disclose all of the elements of all the claims in the patent, in this case the '361 patent.

Q. And what is your understanding of obviousness?

A. So my understanding of obviousness is that obviousness requires that one obtain one or more references, and these references, with some -- if you have more than one you have to show motivation. Would it be sufficient for someone skilled in the state of the art to be able to fulfill all the elements of all the claims in the patent?

Q. What reference did you rely upon to form your opinion regarding invalidity?

A. So I used the Check-Point Firewall 1, its architecture and the administration document.

Q. It's PTX-188.

Is this the document that you are referring to?

A. Yes, it is.

Q. Do you refer to it sometimes as the Check-Point reference or the CP reference?

A. I think usually the CP reference.

Q. Now, was this particular reference cited to the Patent Office during the time that they were applying for a patent

1495

Jaeger - direct

in connection with the '361 patent?

A. Yes, it was.

Q. Was the entire reference cited?

A. No. Just, you probably have the page number, I think it was 135 to 154.

Q. Why don't we show JTX-5.

On the right-hand column, under other publications, could you blow that up, please. Do you see the Check-Point reference here?

A. Yes, I do.

Q. Is it the Check-Point account management client Version, it continues on?

A. Yes, it is.

Q. Let's talk about this Check-Point reference. PTX-188. What is it about?

A. So the Check-Point is a firewall. It looks at individual packets, and so this reference describes how this particular firewall works, to authorize, a Check-Point authorizes both outgoing packets. We talked in the patent, the patent talks about outgoing packets. The Check-Point firewall also authorizes patents that will come from the external network into your network.

And the Check-Point firewall has rules which determine, in this case what the rules do is they describe for a particular entity or group of entities what the

1496

Jaeger - direct

authentication requirements are for that entity to enter the system.

So if Alice made a request, she may correspond to some group of users, and that will match a rule in the firewall. And then based on the authentication requirements in that rule, if Alice can prove that she is really Alice, then she can perform the request specified.

Q. Is it your understanding that this Check-Point reference is prior art to the '361 patent?

A. Yes, it is.

Q. If we could highlight the date, enlarge it, I believe it says September 1998 on the front page?

A. Yes, it does.

Q. Now, how does the Check-Point reference anticipate the elements of Claim '361? Maybe it would be easier if we go through each of the elements. Is the Check-Point reference a system for authorizing client access to a network resource?

A. I was expecting it to be put up. Sorry. Can you repeat it?

Thanks.

Yes, it is.

Q. Does it also have the next element, which reads, A server having at least one directory that can be accessed using a network protocol, said at least one directory being

1497

Jaeger - direct

configured to store information concerning an entity's organization?

A. The Check-Point reference discloses a server for that purpose, yes.

Q. Why don't we turn to PTX-188 at 12633. What does this page tell you with regard to the particular element a server?

A. So this page begins a chapter describing how you can use the Check-Point firewall to leverage the information in a directory server, in this case this LDAP, or lightweight directory access protocol server, this board chart like directory server that we talked about before.

Q. And if we turn three pages later to 12636, what does this tell you about the Check-Point reference in connection with this element we are talking about, a server having one directory?

A. You might want to blow up the firewall. So this London component represents the Check-Point firewall.

Q. When you say London component, what are you referring to?

A. The component with the word London above it.

So in order to determine whether you are going to be authenticated, it may look at an LDAP server. In this case the LDAP server is called BigBen. It's to the right of the London server.

1498

Jaeger - direct

Q. So are these just a few examples that support your opinion that this first element regarding a server found in Claim 1 are found in this Check-Point reference?

A. Yes.

Q. I would like to turn to the next element, G-121, of Claim 1, it starts out, A firewall, and continues all the way to the end of that claim. Do you see that element?

A. Yes, I can.

Q. Does the Check-Point reference disclose this element regarding a firewall?

A. Yes, it does, in my opinion.

Q. Why don't we turn to PTX-188, at 12530.

What does this page tell you in connection with the firewall element?

A. Basically, what this page is telling me, you can picture -- where did it go?

Q. I think this is the right page here.

A. Okay.

I think there was something useful in that other page. But what the --

Q. Why don't I highlight the beginning portion of that page, Under this rule?

A. Okay. Thank you.

So this is saying that a user who is trying to use this particular service called TELLNET, -- this is for

1499

logging into a computer -- is going to be intercepted by the firewall module, which is this entity called London that we took down.

Q. Why don't we turn to 12640 of the same document.

Does this document also support your opinion with regard to the disclosure of the server element in the Check-Point reference?

A. Yes, it does.

Q. Can you point out where exactly in the document?

A. Well, the servers, the LDAP servers are these unfortunately gray and hard to read, especially from this distance, boxes, that the Check-Point firewall calls account units.

Q. Did you cover everything?

A. I was just going to say that each of these account units is an LDAP database. So the idea is, you have part of your company, if you will, part of your user base is captured in each of these account unit directories.

Q. I would like to show you, then, the next page, 12641. Let's highlight Steps 6 through 10. Is there anything on this page that describes to you or supports your opinion regarding this element about the said firewall being operative to authorize a network resource request?

A. So this is describing the process of the firewall using the LDAP directory to authorize such a request, yes.

1500

Q. So does this also further support that the Check-Point reference discloses this entire firewall element that we are talking about in Claim 1?

A. Yes. The whole -- the long element, yes.

Q. How does the comparison work in the Check-Point reference?

A. So -- are you talking about the comparison of what's in the directory to the authorization filter?

Q. That's correct.

A. So what's going to happen, they use the user name Jim in this example.

So what's going to happen is, in the earlier steps, Jim made a network request that was intercepted by the firewall module. So the firewall module had looked in its local database, but it didn't see anything about Jim in its local database.

So it will go out to these account units and see if there is an account unit that knows something about Jim. If there is an account unit that knows something about Jim, it will then, and for the particular request Jim is asking for, it will return a -- I am sorry. If there is an account unit that knows something about Jim, it will return Jim's directory entry from that account unit.

So based on that directory entry, it will then determine what group Jim belongs to. So most of the rules

1501

in Check-Point are written in terms of groups. And so they will determine what group Jim belongs to. And then the LDAP entry also contains information about how to authenticate that Jim is really Jim. So will use that then to determine whether Jim is going to get access through an authorization code.

Q. Let's turn to the next claim, which the Claim 2. G-121.

The additional element here is that one directory is a lightweight directory, access protocol directory. Does the Check-Point reference anticipate Claim 2?

A. Yes, it does.

Q. Does it disclose and describe the lightweight directory access protocol directory?

A. Yes, that is the account units.

Q. Let's turn to the next claim, Claim 3, Wherein said authorization filter is specified using a graphical user interface.

Does the Check-Point reference anticipate Claim 3?

A. Yes, it does. It does provide something for defining its authorization filters.

Q. PX-1288 at 12639.

You found that in the Check-Point reference

1502

itself?

A. Yes, I did.

Q. On this page here?

A. So this, under No. 5, this is showing how you are using, so these account units will define groups that you may not have known about, they may have been at another part of your company or something like this. They are administered perhaps by somebody else. And up above we defined that we will work with that particular group. But this particular gooey here shows how you use the group to write an authorization filter. So basically defining what group you belong to, which is the source, and then there is information about what kind of network request you can perform.

Q. So let's turn them to Claims 4 and 5, which are dependent on Claim 1. Claim 4 has the additional element wherein said authorization filter implements a per-user authentication scheme. And 5 implements a per-service authentication scheme. Is it your opinion that the Check-Point reference anticipates these claims?

A. Yes, it is my opinion that the Check-Point reference does anticipate these two claims.

Q. Why don't we turn to PTX at 12526.

What does the Check-Point reference describe in connection with the per-user and per-service authentication

1503

Jaeger - direct

scheme?

A. So the Check-Point firewall provides actually three schemes, two of which are important to us in this context. One is the first one, called user authentication. And this corresponds to a per-user authentication scheme. And then the third one at the bottom is called session authentication.

So in this case, what they are talking about is a particular session with a particular service.

So you are going to TELLNET. You are going to log into a computer and you will authenticate for the duration of that particular session of that service. These services, you may have to send multiple messages to them in order to use them. This session authentication will support you communicating multiple messages through the firewall to the service. But you only have to authenticate once.

Q. Are there other places that describe this authentication throughout the Check-Point reference? Are there additional references within the Check-Point reference to the different types of authentication?

A. Well, there are major subsections of this document that describe in detail how these particular authentication mechanisms work.

Q. You didn't just rely on this particular page? There is actually a lot more in the reference that discloses --

1504

Jaeger - direct

A. That's correct.

Q. Why don't we turn, then, next, to Claim 7, which has this additional element, wherein said firewall is configured to query multiple directories. What is your opinion regarding Claim 7?

A. My opinion is that that is anticipated by the Check-Point firewall.

Q. Show PTX at 12637.

A. In this picture we have the four gray boxes, the four account units, these are all LDAP directories. The Check-Point is showing us that we can have more than one LDAP directory.

Q. That relates to the firewall being configured according to multiple directories. Is that correct?

A. That's correct.

Q. Let's turn then to Claim 8. Is it your opinion that the Check-Point reference does disclose an authentication method at a firewall?

A. Yes, it is.

Q. Then turning to 8(a), does it disclose this step in 8(a)?

A. Yes, it does. The firewall can intercept requests emanating from inside the network or from outside the network, as we discussed.

Q. We previously discussed some of the pages from the

1505

Jaeger - direct

reference regarding this network request. Is that correct?

A. Yes.

Q. Then turning to 8(b), the element starting Querying, does the Check-Point reference disclose 8(b)?

A. Yes, it does.

So it'll -- as we discussed in the case with Jim, when Jim submits a network request, at least that's I guess an incoming request from the external network, but the idea would be the same. If Jim were inside the network and submitting a request, the firewall would intercept that request and look for information about Jim if it didn't know him in the local database.

Q. And did we talk about some of the pages from the Check-Point reference earlier in connection with this element?

A. Yes, we did.

Q. Let's turn to the next -- actually, why don't we turn to PTX at 12525. Actually, let's go to 12643.

My questions are going to be directed at this particular page regarding the element that the query is based upon authorization filter that is generated based upon a directory schema that is predefined by said entity. What does this particular page tell you in connection with those elements, that element of the claims?

A. So this page and subsequent pages describe the LDAP

1506

Jaeger - direct

schema, that is the structure, the form of the information stored in the LDAP directory for the Check-Point firewall.

Q. Can we show 12641. This one, I have a question in connection with determining -- this is the element, determining based on the results of said query whether the content of at least one or more directories -- it continues on, satisfy the authorization filter.

Looking at Steps 8 through 12, what does that tell you about this reference?

A. So we found an LDAP entry for Jim, from some account unit somewhere. So this LDAP entry has information describing what groups Jim belongs to, and the firewall is going to determine based on these groups what the authentication requirements are for Jim.

I think we will see that in the LDAP scheme later. There will be specific authentication requirements for Jim in order to satisfy gaining entry through the request that he made. So Jim has to prove that he is Jim. And he will use this LDAP entry in order to -- the system will use the LDAP entry in order to describe what Jim has to do to prove that he is Jim so he can submit this request.

So 11 says that the firewall is going to implement the authentication scheme. So it is going to use this information that it found in its LDAP entry in order to determine how to authenticate Jim, and if Jim successfully

1507

authenticates, then the connection will be allowed it says at the bottom.
Q. So based on these pages that we have looked at, turning to Claim 8 again, is it your opinion that all of the elements, that includes (c) and (d) as well are anticipated by the Check-Point reference?
A. Yes, that is my opinion.
Q. What about Claim 9? Is it your opinion that Claim 9 is anticipated?
A. Yes, it is. We use that LDAP data -- directory, I should say. LDAP directory.
Q. So just like Claim 2 it has this element?
A. That's correct.
Q. So we previously discussed the support in the Check-Point reference for the LDAP directory. Correct?
A. That's correct.
Q. Let's turn to Claim 10. Is it your opinion that Claim 10 anticipates?
A. Yes, it is.
Q. Sorry. Let me try that question again. Is it your opinion that the Check-Point reference anticipates Claim 10?
A. '361. Yes.
Q. And this also relates to the graphical user interface like Claim 3. Correct?
A. That's correct.

1508

Q. Basically, based on the information we have looked at already, is it your opinion it anticipates?
A. That's correct.
Q. Turning to Claim 11 and 12, these appear to corresponds to Claims 4 and 5 previously?
A. That's correct.
Q. Based on the same reasons, are these claims anticipated by the Check-Point reference?
A. That is correct.
Q. Then we need to turn to Claim 14. I believe this is also similar to Claim 7?
A. That's right.
Q. And for the same reasons is Claim 14 anticipated by the Check-Point reference?
A. Yes. Claim 14 of '361 is anticipated for the same reasons as Claim 7 of the '361.
Q. We are saying for the same reasons, what we are referring to is just the last new element that is added to the dependent claim?
A. Well, we are saying that it queries multiple databases.
Q. Then for the reasons we had talked about in connection with the dependent claim, for example, in Claim 14, it relies on Claim 8, we are referring to all of those reasons that you discussed in connection with the dependent claim.

1509

Correct?
A. That's right. So, yes, also the reasons that Claim 8 is anticipated.
Q. Let's turn to the last claim of the '361 patent, Claim 15.
Is it your opinion that Claim 15 is anticipated by the Check-Point reference?
A. Yes, it is. My opinion is that it is anticipated as well.
Q. And the preamble, a computer program product for enabling a processor in a computer system to implement an authentication process. Do you see that?
A. Yes, I do.
Q. That particular element is disclosed in the Check-Point reference. Is that correct?
A. Yes, it is.
Q. Turning to the next element, the computer-usable medium. Then it continues onward. Is it your opinion that that particular element is also discussed in the Check-Point reference?
A. Yes. The Check-Point is a computer-usable medium meeting these requirements.
Q. Then turning to the first computer-readable program code element, referring to that entire element, is that also disclosed in this Check-Point reference?

1510

A. Yes, it is.
Q. Is this similar to Element 8(a) that we talked about before?
A. Yes, it is.
Q. So --
A. That is why it has program code, to receive a network request, as we discussed, 8(a).
Q. So based on the same reference that we had looked at earlier?
A. Yes.
Q. This element is anticipated?
A. Yes, it is.
Q. Then turning to the next element, I am not reading the whole thing, it starts out, Second computer-readable program and ends, predefined by said entity, does that element appear in the Check-Point reference?
A. Yes, that element appears also in the Check-Point reference.
Q. Have we discussed all of these, the elements found in that particular element of Claim 15, previously?
A. Yes, we have.
Q. Does that correspond to 8(b)?
A. Yes, it does.
Q. Then let's turn to the next element, which I believe is the third computer readable program code. It finishes

off satisfy said authorization filter. Do you see that reference?

A. Yes, I do.

Q. Was that particular element disclosed in the Check-Point reference?

A. That particular element is also disclosed in the Check-Point reference.

Q. Can we turn to PTX-188 at 12634.

We have looked at this page previously in connection with your discussion of some other claim elements?

A. Yes, we have.

Q. Does this particular page also support your opinion with regard to that element regarding the third computer readable program, as it relates to the passwords?

A. Yes. So, can you bring up the claim again? Sorry. Make sure I get this right.

Q. We are looking at the third computer readable program code?

A. Right. So the account units store these LDAP entries about the individual. So it will store an entry about Jim. And the authorization filter says that Jim has to authenticate, if in that case it does, then the Check-Point firewall will look at the authentication information in Jim's entry in order to determine whether Jim really is Jim.

It's the one thing that you could look at, is Jim's password. That is how we normally authenticate these days. So the Check-Point firewall has that password as saw in the other picture, it actually has a hash of the password, typically.

So what the firewall can do is it can use the value that Jim provided for his password, compute what's called the cryptographic hash. Basically, we don't want to store the passwords in the clear on the directory because if someone, you know, gets in there and sees it, they will have your password. So they will store it in a form that is hard to predict in advance.

But basically what this is is information sufficient for the firewall to determine that you possess that secret, that Jim possesses that secret. So then Jim, if he possesses the secret, he is authenticated, then he can -- the Check-Point system will allow him to enter the network.

Q. Let's just turn to the last element of Claim 15, which reads fourth computer readable program, and it ends said authorization filter is satisfied.

Is it your opinion that that element is disclosed in the Check-Point reference?

A. Yes, it is.

Q. And is this similar to the 8(d) element that we

discussed previously?

A. Yes, it is.

Q. For the same reasons you are basing your opinion?

A. Yes, I am.

Q. So I just want to be clear. Is it your opinion then the Check-Point reference disclosed all the elements of the asserted claims of the '361?

A. It is my opinion that the Check-Point reference asserts all of the elements of this claim, yes.

Q. Is it your opinion that one of ordinary skill in the art would find the '361 patent obvious in light of the Check-Point reference?

A. So it is my opinion that someone skilled in the state of the art would also find the patent obvious given the Check-Point reference.

Q. Now, you say one of ordinary skill in the art. What do you mean by that?

A. So what I mean by that would be either someone with, you know, like a computer science degree, or maybe, you know, about three, four years of professional computer science experience without a degree. That person, additionally, having maybe a couple of years in networking and security. About that level. It would be someone at that level.

Q. For your opinion regarding obviousness of the '361

patent, are you basing it on the same things that we have already discussed previously with regard to the Check-Point reference?

A. Yes, I am.

Q. Now, it's your opinion that the Check-Point reference anticipates and also renders obvious the claim. What is your understanding of the difference between the two?

A. Well, my opinion is that the Check-Point reference anticipates the '361. So, by anticipates, the Check-Point reference discloses all of the elements, it's a single reference and it discloses all of the elements of all of the claims in the '361 patent.

Additionally, I am asserting that the Check-Point reference also renders the '361 patent obvious in the sense that someone of ordinary skill in the state of the art would be able to take this reference and fulfill, understand how to fulfill each of the elements.

MS. KOBIALKA: Your Honor, at this time it might be good to take a break, because we are coming to a good breaking point.

THE COURT: Okay. Let's take our morning break.

(Jury leaves courtroom at 10:55 a.m.)

(Recess taken.)

THE COURT: Counsel, we are going to have to order lunch for the jury. How are we doing?

instructions that you read when I get back to the office. They have not been filed.

THE COURT: I have the originals. We will scan them in.

MR. ROVNER: That is fine.

THE COURT: Thank you, counsel.

(Court recessed.)

- - -

Reporter: Kevin Maurer

EXHIBIT 3
PART 1

Check Point FireWall-1™ Architecture and Administration

Version 4.0

Part No.: 71300001400
September 1998

CHECK POINT™
Software Technologies Ltd.

FireWall-1

Plaintiff's Trial Exhibit
PTX-188
Case No. 06-369 GMS

RESTRICTED RIGHTS LEGEND:

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

TRADEMARKS:

Check Point, the Check Point logo, FireWall-1, FireWall-First!, FloodGate-1, INSPECT, IQ Engine, Open Security Manager, OPSEC, SecuRemote, UAP, VPN-1 and ConnectControl are registered trademarks or trademarks of Check Point Software Technologies Ltd.

All other product names mentioned herein are trademarks of their respective owners.

The products described in this document are protected by U.S. patent no. 5,606,668 and may be protected by other U.S. patents, foreign patents, or pending applications.

THIRD PARTIES:

Entrust is a registered trademark of Entrust Technologies, Inc. in the United States and other countries. Entrust's logos and Entrust product and service names are also trademarks of Entrust Technologies, Inc. Entrust Technologies Limited is a wholly owned subsidiary of Entrust Technologies, Inc. FireWall-1 and SecuRemote incorporate certificate management technology from Entrust.

The following statements refer to those portions of the software copyrighted by University of Michigan.

RealSecure, SAFEsuite, Intranet Scanner, Internet Scanner, Firewall Scanner, and Web Scanner are trademarks or registered trademarks of Internet Security Systems, Inc.

The following statements refer to those portions of the software copyrighted by Carnegie Mellon University.

Check Point Software Technologies Ltd.

International Headquarters:
3A Jabotinsky Street
Ramat Gan 52520, Israel
Tel: 972-3-753 4555
Fax: 972-3-575 9256

e-mail: info@CheckPoint.com

U.S. Headquarters:
Three Lagoon Drive, Suite 400
Redwood City, CA 94065
Tel: 800-429-4391 ; (650) 628-2000
Fax: (650) 654-4233

http://www.checkpoint.com

August 9, 1998

Please direct all comments regarding this publication to techwriters@checkpoint.com.

Please Recycle

Adobe PostScript

Contents

Figures

Tables

Preface

Scope

The FireWall-1 User Guide describes all CheckPoint FireWall-1 products, and consists of the following books:

Getting Started with FireWall-1

This book introduces FireWall-1 and describes the FireWall-1 installation process.

Managing FireWall-1 Using the OpenLook GUI

This book describes how to manage FireWall-1 using the OpenLook Graphical User Interface (GUI).

Managing FireWall-1 Using the Windows GUI

This book describes how to manage FireWall-1 using the Microsoft Windows Graphical User Interface (GUI).

FireWall-1 Architecture and Administration

This book is the technical reference to FireWall-1 features, including authentication and address translation. In addition, chapters on troubleshooting and Frequently Asked Questions (FAQ) are included.

Virtual Private Networking with FireWall-1

This book describes how to establish a Virtual Private Network using FireWall-1.

Account Management Client

This book describes how to install and use the Check Point Account Management Client.

Who Should Use this User Guide

This User Guide is written for system administrators who are responsible for maintaining network security.

It assumes you have a basic understanding and a working knowledge of:

- system administration
- the Unix or Windows operating system
- the OpenLook or Windows GUI
- Internet protocols (IP)

Summary of Contents

Chapter 1, "Authentication," describes the FireWall-1 Authentication features.

Chapter 2, "Security Servers," describes the FireWall-1 Security Servers, which implement Authentication and Content Security.

Chapter 3, "Content Security, describes the FireWall-1 Content Security feature.

Chapter 4, "Account Management," describes the FireWall-1 Account Management feature, including LDAP servers and Account Units.

Chapter 5, "Network Address Translation," describes the FireWall-1 Address Translation feature.

Chapter 6, "Routers and Embedded Systems," describes how FireWall-1 enforces a Security Policy on routers and embedded systems.

Chapter 7, "Management Server," describes the Client/Server implementation of the Management Module.

Chapter 8, "Active Network Management," describes FireWall-1 that enable system administrators to better manage their networks in real-time.

Chapter 10, "Command Line Interface," describes the FireWall-1 command line interface, an alternative to the Graphical User Interface.

Chapter 11, "INSPECT," describes the FireWall-1 Inspection Script language.

Chapter 12, "Miscellaneous Security Issues," discusses various security issues, including the FireWall-1 SYN attack defense.

Chapter 13, "Troubleshooting," describes common FireWall-1 problems and their solutions.

Chapter 14, "FAQ (Frequently Asked Questions)," is a compilation of questions asked by FireWall-1 users, and the answers to the questions.

Chapter 15, "Directories and Files," lists the FireWall-1 directory structure and the files in each directory.

Chapter 16, "Services," briefly describes many of the communications services supported by FireWall-1.

Chapter 17, "FireWall-1 – Windows Interaction," describes interaction between FireWall-1 and Windows NT, including performance monitoring and the Windows Event Viewer.

Appendix A, "Glossary" gives brief definitions of some FireWall-1 concepts and terminology.

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`

`AaBbCc123`	What you type, when contrasted with on-screen computer output	`machine_name% su`
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, Korn shell and DOS.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	$
Bourne shell and Korn shell superuser prompt	#
DOS	current-directory>

Network Topology Examples

Network topology examples usually show a gateway's name as a city name (for example, Paris or London) and the names of hosts behind each gateway as names of popular sites in those cities (for example, Eiffel and BigBen).

CHAPTER 1

Authentication

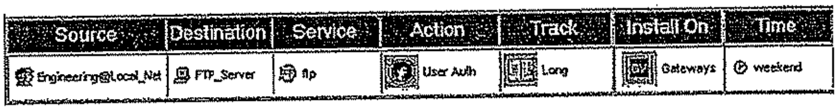
In This Chapter

Note – This chapter uses Windows GUI names for all screen examples. Examples in this chapter are intended to illustrate the use of authentication features, and should not serve as typical configurations or rules.

Overview

FireWall-1 Authentication

FireWall-1's Authentication feature enables you to control security by allowing some users access while disallowing others. Authentication is specified in a rule's **Action** field, so you can be very flexible in the way you combine Authentication with the other fields in a rule. For example, you can allow a group of users at a host or group of hosts to use specific services on specific servers at a given time of day.



Source	Destination	Service	Action	Track	Install On	Time
Engineering@Local_Net	FTP_Server	ftp	User Auth	Long	Gateways	weekend

FIGURE 1-1 Authentication Rule

The rule shown in FIGURE 1-1 on page 27 specifies that users in the group "Engineering" initiating an FTP connection from the local network to the FTP Server on weekends will be authenticated before the connection is allowed.

Three Kinds of Authentication

There are three kinds of Authentication:

- User Authentication

  User Authentication grants access on a per user basis. This method can only be used for TELNET, FTP, RLOGIN and HTTP, and requires a separate authentication for each connection. It is secure (because the authentication is valid only for one connection), but intrusive (because each connection requires another authentication).

- Client Authentication

  Client Authentication grants access on a per host basis. Client Authentication allows connections from a specific IP address after successful authentication. It can be used for any number of connections, for any service and the authentication is valid for the length of time defined by the administrator. It is less secure than User Authentication (because it allows any user access from the IP address or host) but is also less intrusive. It is best used when the client is a single-user machine, such as a PC.

- Session Authentication

  Session Authentication is like User Authentication in that it requires an authentication procedure for each connection, but unlike User Authentication, it can be used with any service. It is secure but requires authentication for each connection. It also requires a Session Authentication agent running on the client or another machine in the network.

Transparent Authentication

Authentication is considered transparent when a user doesn't have to explicitly connect to the FireWalled gateway to perform the authentication before continuing to the destination. The connection attempt (for example, when the user issues the telnet command) is intercepted by FireWall-1 on the gateway and the authentication procedure is activated. If the authentication is successful, and the connection is allowed by the rule, the connection proceeds to the destination.

Comparison of Authentication Types

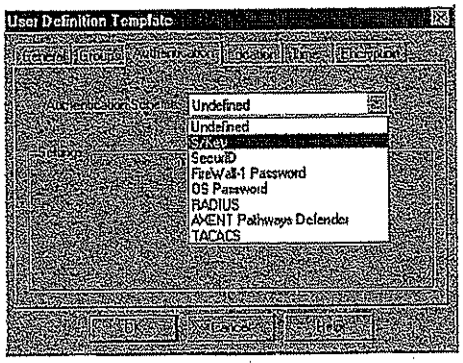
TABLE 1-1 compares the features of the three FireWall-1 Authentication types.

TABLE 1-1 Comparison of Authentication Types

	User Authentication	Client Authentication	Session Authentication
which services	TELNET, FTP, RLOGIN, HTTP	all services	all services
authentication is performed once per ...	session	IP address	session
use when you want a user to ...	authenticate each time he or she uses one of the supported services	authenticate once, and then be able to use any service until logging off	authenticate each time he or she uses *any* service (this requires a Session Authentication Agent running on the client or another machine in the network)

How A User Authenticates

A user authenticates himself or herself by proving his or her identity according to the scheme specified under **Authentication Scheme** in the **Authentication** tab of his or her **User Properties** window.



Authentication Scheme: Undefined

FIGURE 1-2 User Properties window — Authentication tab

Authentication Schemes

FireWall-1 supports the following Authentication schemes:

- **Undefined** — No authentication is performed and access is always denied.
- **S/Key** — The user is challenged to enter the value of requested S/Key iteration.

  For an explanation of how to define S/Key authentication, see Chapter 3, "User Management" in *Managing FireWall-1 Using the OpenLook GUI* or *Managing FireWall-1 Using the Windows GUI*.
- **SecurID** — The user is challenged to enter the number displayed on the Security Dynamics SecurID card.
- **OS Password** — The user is challenged to enter his or her OS password.
- **FireWall-1 Password** — The user is challenged to enter his or her FireWall-1 password on the gateway.

  The advantage of a FireWall-1 password over the OS password is that the user does not require an OS account on the gateway to use a FireWall-1 password.
- **RADIUS** — The user is challenged for the response, as defined by the RADIUS server.

  For information about defining RADIUS servers, see Chapter 5, "Server Objects" of *Managing FireWall-1 Using the OpenLook GUI* or *Managing FireWall-1 Using the Windows GUI*.
- **AXENT Pathways** — The user is challenged for the response, as defined by the AXENT Pathways server.

  In the Windows GUI, the Axent Pathways server is defined as a Server object. For more information, see Chapter 5, "Server Objects" in *Managing FireWall-1 using the Windows GUI*.

  In the OpenLook GUI, the Axent Pathways server is defined in the **Control Properties — Authentication** window. For more information, see Chapter 8, "Control Properties" in *Managing FireWall-1 Using the OpenLook GUI*.
- **TACACS** — The user is challenged for the response, as defined by the TACACS or TACACS+ server.

  For information about defining TACACS servers, see Chapter 5, "Server Objects" of *Managing FireWall-1 Using the OpenLook GUI* or *Managing FireWall-1 Using the Windows GUI*.

A user can have different passwords on different gateways (for example, if the Authentication Scheme is OS Password), but only one Authentication scheme for all gateways.

User Authentication

In This Section

User Authentication — Overview

User Authentication is provided by the TELNET, FTP, HTTP and RLOGIN FireWall-1 Security Servers on the gateway. When a rule specifies User Authentication, the corresponding FireWall-1 Security Server is invoked in order to mediate the connection.

Consider the following configuration and rule:

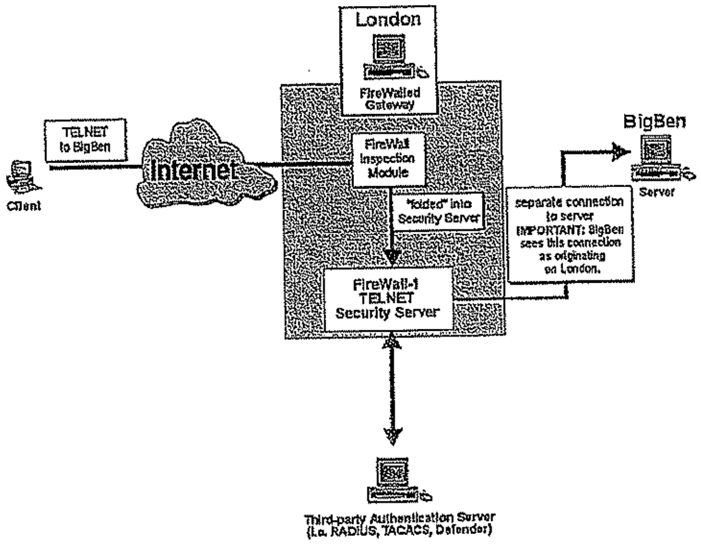

London
FireWalled Gateway
TELNET to BigBen
Client
Internet
FireWall Inspection Module
"folded" into Security Server
FireWall-1 TELNET Security Server
BigBen
Server
separate connection to server IMPORTANT: BigBen sees this connection as originating on London.
Third-party Authentication Server (i.e. RADIUS, TACACS, Defender)

FIGURE 1-3 A connection mediated by the TELNET Security Server

Source	Destination	Services	Action	Track	Install On
All_Users@Any	BigBen	telnet	UserAuth	Long Log	Gateways

Under this rule, a user who TELNETs to BigBen is intercepted by the FireWall Module on the gateway London. The FireWall Module diverts the connection to the TELNET Security Server on the gateway, even though the user did not explicitly connect to the gateway.

The Security Server then authenticates the user, in accordance with the Authentication scheme defined for the user in the **Authentication** tab of the **User Properties** window.

> If no Authentication scheme is specified for a user, the user will be denied access.
>
> If an external Authentication scheme is specified (i.e. RADIUS, TACACS, or Axent Defender) the Security Server queries the appropriate third-party server regarding the user's permissions. The third-party server returns the appropriate data to the Security Server.

If authentication is successful, the TELNET Security Server opens a separate connection to the target server, in this case, BigBen. Altogether, there are two connections, one to the Security Server on the gateway, and another from the Security Server to the final destination. The final destination server sees the connection as originating from the gateway, not the client. Authentication is transparent — the user TELNETs to BigBen, the target server, and not to the gateway.

Note – FireWall-1 also supports non-transparent Authentication for TELNET, RLOGIN, HTTP and FTP. In non-transparent Authentication, the user must first connect directly to the gateway and authenticate before being allowed to continue to the target host. For more information see "Non-Transparent User Authentication" on page 42.

User authentication rules allow users if they authenticate successfully, but do not necessarily reject the connection if the user fails authentication. In addition, the fact that a user successfully authenticates does not necessarily mean that there is a rule that allows that user access. This is because the authenticating Security Server first checks if the connection can be allowed by a rule which does not require authentication. For more information, see "The 'Insufficient Information' Problem" on page 104 of Chapter 2, "Security Servers."

User Authentication — Deployment

This section describes a deployment example for User Authentication. A deployment example consists of:

- an example network configuration
- what the Security Administrator must define in the FireWall-1 Rule Base
- what a user must do to authenticate

This example is not intended as a set of step by step instructions, but rather to illustrate how and where different components of User Authentication are configured in the FireWall-1 GUI.

Example Configuration

FIGURE 1-4 depicts an example configuration in which London, the FireWalled gateway, protects a local network and a DMZ.

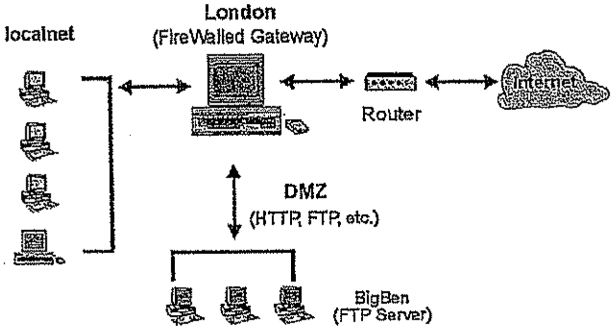
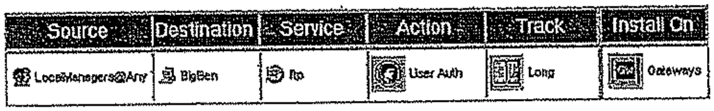


FIGURE 1-4 Example configuration

The Security Administrator for this configuration may want to allow only localnet managers access to files on BigBen, the FTP server. The following rule allows a user group (**LocalManagers**) to access the FTP server after successful User Authentication.



Source	Destination	Service	Action	Track	Install On
LocalManagers@Any	BigBen	ftp	User Auth	Long	Gateways

FIGURE 1-5 Example User Authentication Rule

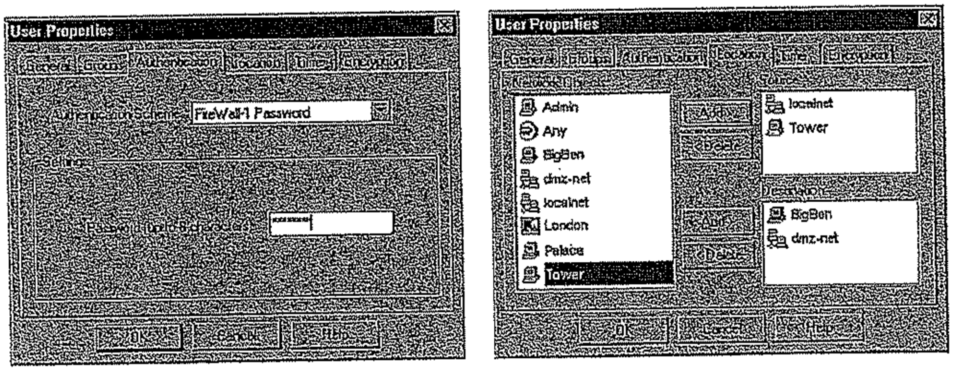
Defining User Authentication

In order to implement User Authentication for this configuration and rule, you must define the following:

- the permitted user group
- authentication schemes supported by the gateway
- tracking and timeout parameters
- User Authentication rule properties

Defining User Properties

In a User Authentication rule, the **Source** must be a user group (i.e., **allusers@localnet**). You must first define the permitted users, their authentication scheme or schemes and the network objects from which each user is allowed access. These properties are defined in the tabs of the **User Properties** window.



Authentication Scheme: FireWall-1 Password

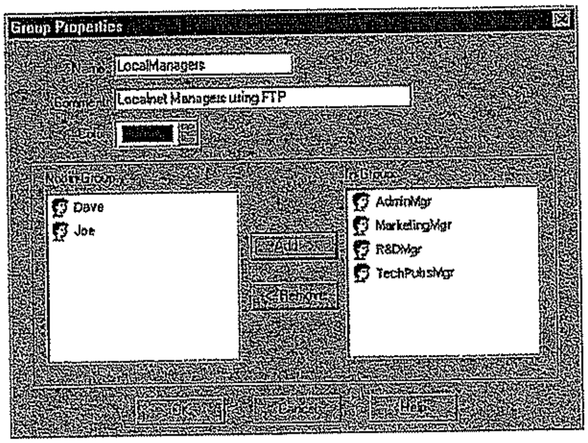
Admin
Any
BigBen
dmz-net
localnet
London
Palace
Tower

localnet
Tower

BigBen
dmz-net

FIGURE 1-6 User Properties window - Authentication tab and Location tab

You must next define a user group called "LocalManagers" consisting of the users who are allowed access.



Name: LocalManagers

Comment: Localnet Managers using FTP

Dave
Joe

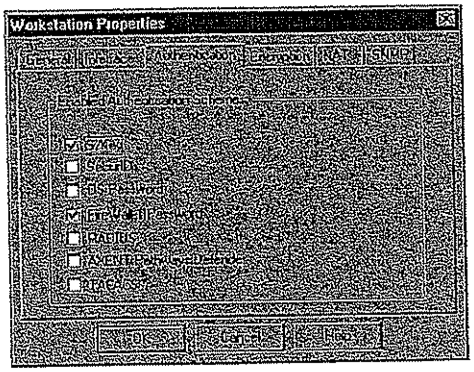
AdminMgr
MarketingMgr
R&DMgr
TechPubsMgr

FIGURE 1-7 Group Properties window - LocalManagers group

For more information on defining users and user groups, see Chapter 3, "User Management" of *Managing FireWall-1 Using the Windows GUI* or *Managing FireWall-1 Using the OpenLook GUI*.

Defining the Gateway's Authentication Schemes

The gateway must support the same authentication schemes you defined for your users. For example, if some users in the group "LocalManagers" are using a FireWall-1 password, and others are using S/Key authentication, you must make sure the gateway supports both the FireWall-1 Password and S/Key authentication schemes. The gateway's authentication schemes are defined in the **Authentication** tab of the **Workstation Properties** window.



Workstation Properties

Enabled Authentication Schemes

☑ S/Key
□ SecurID
□ OS Password
☑ FireWall-1 Password
□ RADIUS
□ AXENT Pathways Defender
□ TACACS

FIGURE 1-8 Workstation Properties window — Authentication tab

Tracking and Timeout Parameters

You must specify tracking for failed authentication attempts and timeout parameters in the **Authentication** tab of the **Properties Setup** window. These parameters apply to all rules.

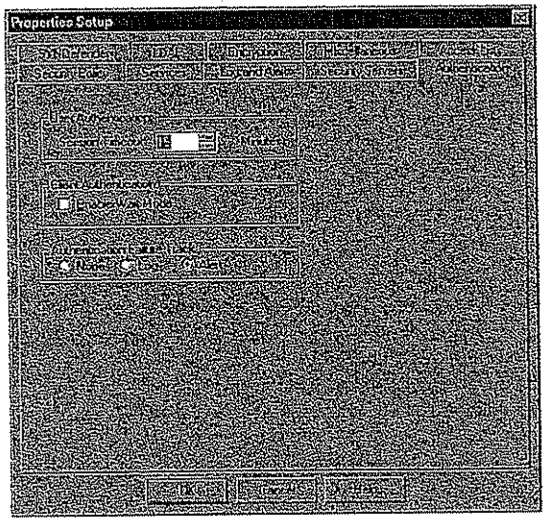


FIGURE 1-9 Properties Setup window — Authentication tab

Note – This window also specifies Client Authentication parameters. For more information, see "Client Authentication" on page 75.

User Authentication: Session Timeout (minutes) — the session will time out if there is no activity (see TABLE 1-2 for the meaning of the term "no activity") for this time period.

This applies to the FTP, TELNET, and RLOGIN Security Servers.

TABLE 1-2 Meaning of "No Activity"

service	the term "no activity" means
FTP	no data transferred
TELNET	no keyboard (or mouse) activity
RLOGIN	no keyboard (or mouse) activity

Session Timeout has a different meaning for HTTP. Users of one-time passwords will not have to reauthenticate for each request during this time period. Each successful access resets the timer to zero.

> Because each connection requires authentication, FireWall-1 extends the validity of one-time passwords for this period. In this way, users of HTTP do not have to generate a new password and reauthenticate for each connection.

Authentication Failure Track — the action to take if Authentication fails (applies to all Authentication rules)

- **None** — no tracking
- **Log** — Long Log
- **Alert** — the **User Authentication Alert Command** in the **Log and Alert** tab of the **Properties Setup** window.

The tracking for a successful Authentication attempt is determined by the **Track** field of the enabling rule. If Authentication is successful then:

- If access is allowed, the **Track** specified in the rule which allows the access is applied.
- If access is denied, the **Track** specified in the rule which denies the access is applied.

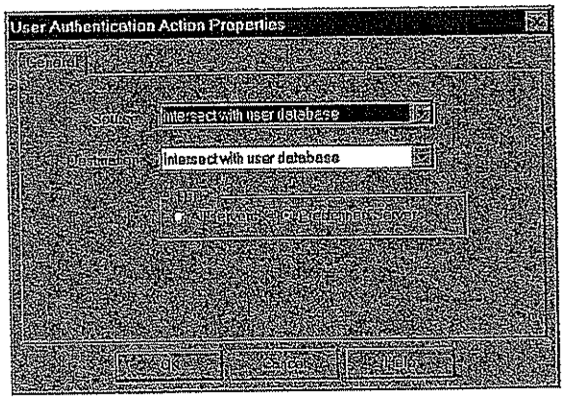
The fact that a user successfully authenticates does not necessarily mean that there is a rule that allows that user access. For more information, see "The 'Insufficient Information' Problem" on page 104 of Chapter 2, "Security Servers."

For example, the rule depicted in FIGURE 1-5 on page 33 specifies logging in Long Log format. If authentication is successful, and access is allowed by this rule, the Long Log format is applied. A failed authentication attempt (for example, using an unauthorized password) is tracked according to the entry under **Authentication Failure Track** in the **Authentication** tab of the **Properties Setup** window.

User Authentication Rule Properties

You must also define the User Authentication properties of the enabling rule. The **User Authentication Action Properties** window specifies the parameters that apply to an individual rule. You can use this window to restrict user access to and from specific network objects.

To display the **User Authentication Action Properties** window, double-click on the rule's **Action**.



Source: intersect with user database

Destination: intersect with user database

● All Servers ○ Preferred Server

FIGURE 1-10 User Authentication Action Properties window

Source — Reconcile **Source** in the rule with the allowed **Sources** in the **Location** tab of the **User Properties** window.

The allowed **Sources** in the **Location** tab of the **User Properties** window may specify that the user to whom this rule is being applied is not allowed access from the source address, while the rule may allow access. This field indicates how to resolve this conflict.

- Choose **Intersect with User Database** to apply the intersection of the access privileges specified in the rule and in the **User Properties** window.
- Choose **Ignore User Database** to allow access according to the **Source** specified in the rule.

Destination — Reconcile **Destination** in the rule with the allowed **Destinations** specified in the **Location** tab of the **User Properties** window.

The **Allowed Destinations** field in the **User Properties** window may specify that the user to whom this rule is being applied is not allowed access to the destination address, while the rule may allow access. This field indicates how to resolve this conflict.

- Choose **Intersect with User Database** to apply the intersection of the access privileges specified in the rule and in the **User Properties** window.
- Choose **Ignore User Database** to allow access according to the **Destination** specified in the rule.

Example

Suppose the **Location** tab of a user's **User Properties** window (FIGURE 1-6 on page 34) lists the network objects **Tower** and **localnet** under **Source**. This means that the user's properties allow access from Tower and localnet TABLE 1-3 summarizes various access possibilities.

TABLE 1-3 Access Possibilities

rule's Source allows access from ...	Source is Intersect with User Database	Source is Ignore User Database
Tower	The user is allowed access only from **Tower**, because only **Tower** is in both **Sources** in the **Location** tab and in the rule's **Source**.	The user is allowed access only from **Tower** because **Tower** is in the rule's **Source**.
Thames	The user is denied access, because there is no network object that is in both **Sources** in the **Location** tab and the rule's **Source**.	The user is allowed access only from **Thames**, because only **Thames** is in the rule's **Source**.

Ignore User Database allows the administrator to grant access privileges to a user without regard to the user's IP address. For example, if a user is temporarily away from the office and logging in from a different host, the administrator may wish to allow that user access regardless of the network objects listed under the allowed **Source** specified in the **Location** tab (see FIGURE 1-6 on page 34) of that user's **Properties** window. The administrator can allow the user to work on the local network without extending access privileges to all users on that host.

HTTP — This option allows you to restrict access to specific HTTP servers. (For more information, see "HTTP Servers List (Security Servers tab)" on page 46).

How the User Authenticates

In the example configuration depicted in FIGURE 1-4 on page 33, a user in the "LocalManager" group using a FireWall-1 authentication scheme must request an FTP session on BigBen. The user must provide the following information:

- user name on the gateway
- authentication data (password) on the gateway
- user name on the target host
- authentication data (password) on the target host

Example - FTP Using O/S Password

```
tower # ftp bigben
Connected to london.
220 london CheckPoint FireWall-1 secure ftp server running on London
Name (bigben:jim): jimb
331-aftpd: FireWall-1 password: you can use password@FW-1-password
Password: <Unix password on bigben>@<FireWall-1 password>
230-aftpd: User jimb authenticated by FireWall-1 authentication.
230-aftpd: Connected to bigben. Logging in...
230-aftpd: 220 bigben ftp server (UNIX(r) System V Release 4.0) ready.
230-aftpd: 331 Password required for jimb.
230-aftpd: 220 User jimb logged in.
```

In the above example, the user requested an FTP session on BigBen, the target server. The steps involved were as follows:

1 The user requested an FTP session on BigBen, the target server, by typing `ftp bigben`.

The connection was intercepted by the FireWall-1 FTP Security Server on London, the gateway. The Security Server prompted the user for his name on the gateway.

2 The user typed `jimb`, meaning that his user name on the target (BigBen) is jimb and his user name on the gateway is jimb. For more information how user names are entered, see "Entering User Names — FTP."

3 The FireWall-1 Security Server challenged the user for his FireWall-1 password, in accordance with the Authentication scheme specified for the user in the **Authentication** tab of the **User Properties** window.

4 The user correctly entered the FireWall-1 password and was connected to BigBen, the target server. For more information on how the user enters his or her password, see "Parsing the Password String — FTP" on page 41.

The FireWall-1 FTP Security Server on London supplied FTP on BigBen with the user's password, so the user did not have to enter it again.

Entering User Names — FTP

For FTP, the user name can be entered in one of three ways:

- *user_name* — this is the user name on both the gateway and the target server.
- *user_name@host* — where *user_name* is the user name on both the gateway and the target server, and *host* is the name of the target server.

- *user_name@fw_user_name@host* — where *user_name* is the user name on the target server, *fw_user_name* is the user's name on the gateway, and *host* is the name of the target server. This is used when the user has different names on the gateway and the target server.

Parsing the Password String — FTP

For FTP, FireWall-1 parses the password string as follows:

- Whatever is to the left of the last `@` is interpreted as the password to the FTP server.
- Whatever is to the right of the last `@` is interpreted as the User Authentication password.

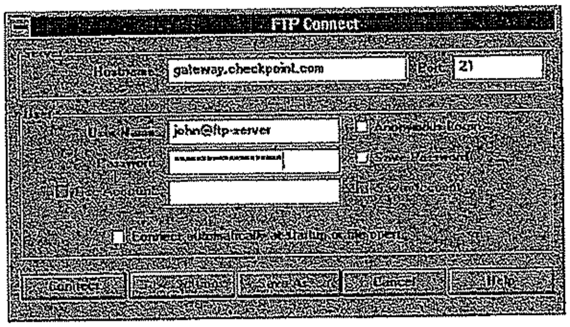
For example, if the user types `jimb@BigBen.com@secret`, then `jimb@BigBen.com` is the FTP server password, and `secret` is the FireWall-1 User Authentication password.

This feature is important when using anonymous FTP and entering an email address as the password.

If the user wishes to postpone entering the FireWall-1 Authentication password, then he or she should type the FTP server password and add `@` at the end. The user will be prompted for the FireWall-1 Authentication password later.

Additional Features — User Authentication

Authentication Using GUI Clients



FTP Connect

Hostname: gateway.checkpoint.com

Port: 21

User Name: john@ftp-server

FIGURE 1-11 GUI FTP Authentication

An FTP or TELNET user with a GUI (see FIGURE 1-11 on page 41) must specify his or her user name and password as follows:

User Name

dst_user_name [@auth_user_name] @ *dst*

where:

- *dst_user_name* is the user name on the destination host
- *auth_user_name* is the user name on the gateway
- *dst* is the name of the destination host

Password

dst_password [@auth_password]

where:

- *dst_password* is the password on the destination host
- *auth_password* is the password on the gateway

Non-Transparent User Authentication

In This Section

Non-transparent User Authentication can be implemented for user authenticated services (FTP, HTTP, RLOGIN, TELNET). Under non-transparent user authentication, a user working with one of these services must first start a session for that service on the gateway. After successful authentication, FireWall-1 opens a connection to the "true" destination. This method is known as non-transparent because the user does not directly request the target host, but must explicitly connect the gateway first.

Non transparent User Authentication was implemented previous version of FireWall-1 as follows:

- In Version 2.1 and earlier versions, FireWall-1 implemented non transparent User Authentication.
- In Version 3.0, non-transparent User Authentication could be implemented to enable a pre-Version 3.0 Management Module to download the Security Policy to a Version 3.0 FireWall Module.

Note – Transparent User Authentication is the default action for FireWall-1 version 3.0 and higher.

Non–Transparent User Authentication — Example

Consider the following network configuration.

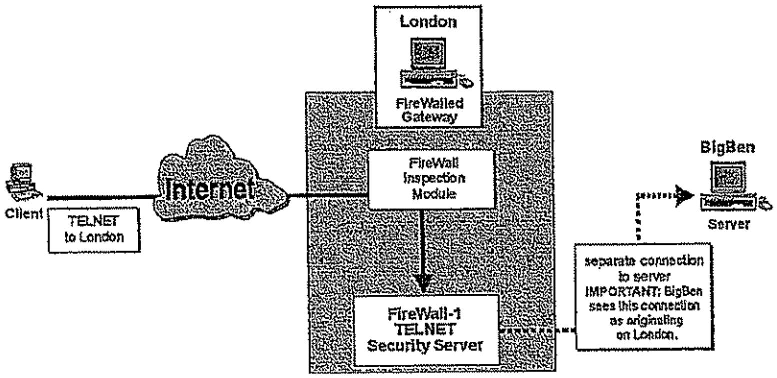

London
FireWalled Gateway
BigBen
FireWall Inspection Module
Internet
Client
TELNET to London
Server
FireWall-1 TELNET Security Server
separate connection to server IMPORTANT: BigBen sees this connection as originating on London.

FIGURE 1-12 Non–Transparent User Authentication.

In the above configuration, the user who wants to begin a TELNET session on Big Ben must first TELNET to London (the gateway) where the FireWall-1 TELNET Security Server is installed on the TELNET port in place of the standard TELNET daemon. The user must then specify BigBen as the ultimate destination. The FireWall-1 TELNET Security Server authenticates the user, and if the authentication is successful, opens a connection to BigBen, the "true" destination.

The user must provide the following information:

- user name on the gateway
- authentication data (password) on the gateway
- name of the target host — **this is the major difference between transparent and non-transparent authentication**
- user name on the target host
- authentication data (password) on the target host

Enabling Non-Transparent User Authentication

Non transparent user authentication is controlled by the `prompt_for_destination` parameter in the file `objects.C`.

The `prompt_for_destination` parameter determines what to do when the user connects directly to the FireWalled gateway. In this case, there are two possibilities:

- `prompt_for_destination` is `false`. This is the default value for FireWall-1 version 3.0 and higher. If `prompt_for_destination` is `false`, FireWall-1 implements transparent user authentication. The FireWall Module assumes the user's "true" destination is the FireWalled machine, and does not prompt the user for the "true" destination.

- `prompt_for_destination` is `true`

  This is the value for implementing non transparent authentication. If the user requests the gateway as the destination, the FireWall Module assumes the user's "true" destination is some other server, and prompts the user for the "true" destination.

  Non-transparent authentication will be implemented only if the user first requests the gateway. If the user requests a host behind the gateway, FireWall-1 will implement non-transparent authentication, regardless of the setting for `prompt_for_destination`.

Note – After making changes to `objects.C`, you must download the database from the Management Station to the FireWall in order for the changes to take effect. For more information, see, "fw dbload" on page 264.

User Authentication and the HTTP Security Server

In This Section

Overview

The FireWall-1 HTTP Security Server provides a mechanism for authenticating users of HTTP services. A FireWall-1 HTTP Security Server on the gateway can protect any number of HTTP servers behind the gateway, and authenticate users accessing HTTP or HTTPS (HTTP encrypted by SSL).

The HTTP Security Server is invoked when a rule's action specifies **User Authentication**. The rule's **Service** can specify either the HTTP service or a Resource. This section describes the behavior of the HTTP Security Server for User Authentication rules. For more information on how the HTTP Security Server handles Resources, see Chapter 3 "Content Security" of this book.

HTTP Security Server Configuration

HTTP Security Server parameters are defined in the **Security Servers** tab and the **Authentication** tab of the **Properties** setup window.

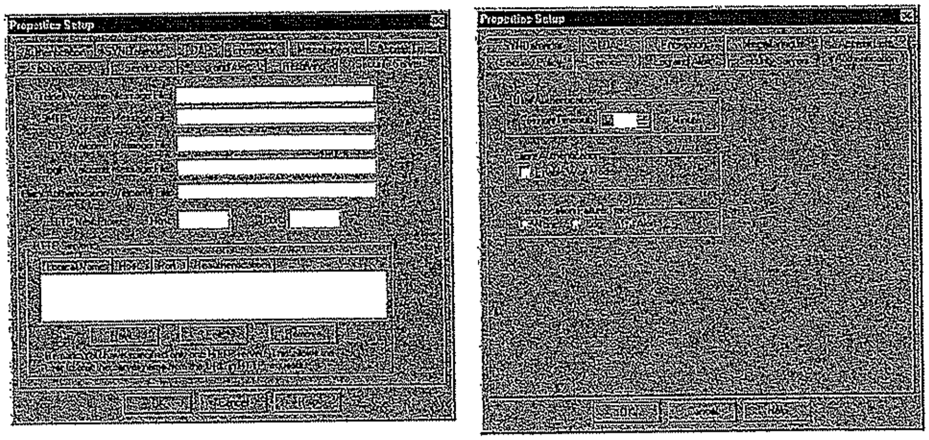


FIGURE 1-13 Properties Setup window — Security Servers and Authentication tabs

The fields relating to the HTTP Security Server are explained below.

- **Session Timeout (Authentication** tab) — For HTTP, this applies to one-time passwords only. Users of one-time passwords to not have to reauthenticate during this time period. The HTTP Security Server extends the validity of a one-time password for this time period so the user does not have to generate a new password and reauthenticate for each connection. Each successful access resets the timer to zero. For more information, see "HTTP Security Server — Security Considerations" on page 57.
- **HTTP Next Proxy (Security Policy** tab) — the **Host** name and **Port** number of the HTTP proxy behind the FireWall-1 HTTP Security Server (if there is one)

  This option is useful if internal users have defined the HTTP Security Server as the proxy to their Web browsers (see "The HTTP Security Server as an HTTP Proxy" on page 50). The FireWall-1 HTTP Security Server does not cache pages used by its client (the browser). If you wish to provide caching for HTTP users, you can put an HTTP proxy behind the FireWall-1 HTTP Security Server.

  Changing the **HTTP Next Proxy** fields takes effect after the FireWall-1 database is downloaded to the authenticating gateway, or after the Security Policy is re-installed.

HTTP Servers List (Security Servers tab)

The **HTTP Servers** list in **Security Servers** tab of the **Properties Setup** specifies the host names and port numbers of HTTP servers.

Defining HTTP Servers allows you to restrict incoming HTTP. You can control access to specific ports on specific hosts. You can also specify whether users must reauthenticate when accessing a specific server.

If you are implementing Non-transparent Authentication, then the **HTTP Servers** list provides a list of hosts and port numbers to which the HTTP Security Server can direct connections from the gateway. For more information, see "HTTP Security Server and Non-Transparent Authentication" on page 58.

If you change the location of an HTTP server (i.e., you install it on a new host or port), you must update the **HTTP Servers** list. For more information, see "Controlling Access to HTTP — User Authentication Rules" on page 48.

Configuring HTTP Servers

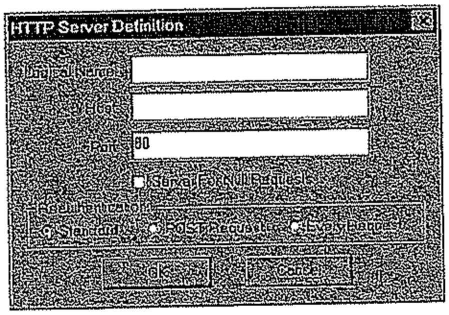
To add a new server, click on **New**. The **HTTP Server Definition** window (FIGURE 1-14) is then displayed.

To delete a server from the list, select it and click on **Remove**. More than one server can be selected at a time.

To modify a server's host or port number, click on the server's logical name in the **HTTP Servers** list and click on **Edit**. The server's details are then displayed in the **HTTP Server Definition** window (FIGURE 1-14).

HTTP Server Definition window

This window defines an HTTP server.



Port: 80

FIGURE 1-14 HTTP Server Definition window

The fields in the **HTTP Server Definition** window are explained below:

Logical Name — the server's logical name

Host — the host on which the server runs

Port — the port number on the host

Server For Null Requests — this option is relevant only if Non-transparent Authentication is enabled. For more information, see "Configuring a Server for Null Requests" on page 60.

Reauthentication Options

Reauthentication options define whether a user must reauthenticate every time the HTTP server is accessed. Reauthentication options apply only when **Predefined Servers** is specified under **HTTP** in a rule's **User Authentication Action Properties** window (FIGURE 1-10 on page 38).

Select one of the following options:

Standard Authentication — The user will not be required to enter a password again during the authorization period (as specified in the **Session Timeout** field in the **Authentication** tab of the **Properties Setup** window). Each successful access resets the timer to zero.

Reauthentication for POST Requests — Every request sent by the client which may change the server's configuration or data requires the user to enter a new password.

If the password is not a one-time password, this option has no effect.

Reauthentication for Every Request — every request for a connection requires the user to enter a new password

If the password is not a one-time password, this option has no effect.

This option is useful when access to some pages must be severely restricted. It is recommended that pages such as these be handled by a separate server.

The Reauthentication status of each HTTP Server is displayed under **Reauthentication** in the **HTTP Servers** list.

Controlling Access to HTTP — User Authentication Rules

Restricting Incoming HTTP

If you wish to restrict access to internal HTTP services by external users, proceed as follows:

1. Define a rule similar to the following:

Source	Destination	Services	Action	Track	Install On
All_Users@Any	localnet	http	UserAuth	Long Log	Gateways

The above rule requires external users to authenticate before accessing HTTP services in the local network. Incoming HTTP is intercepted by the HTTP Security Server on the gateway.

2. In the rule's **User Authentication Action Properties** window (FIGURE 1-10 on page 38), choose **Predefined Servers** under **HTTP**.

This restricts external access to the HTTP servers listed in the **Security Servers** tab of the **Properties Setup** window. This will also activate the **Reauthentication** options specified for the HTTP servers. For more information, see "HTTP Servers List (Security Servers tab)" on page 46.

Restricting Internal Users' Access to External HTTP

To restrict internal users' access to external HTTP, proceed as follows.

1. Define a rule similar to the following:

Source	Destination	Services	Action	Track	Install On
All_Users@localnet	any	http	UserAuth	Long Log	Gateways

This rule allows internal users to use external HTTP if they first authenticate themselves on the gateway.

2. In the rule's **User Authentication Action Properties** window (FIGURE 1-10 on page 38), choose **All Servers** from the HTTP options. This allows the connection to continue on to any port.

If **All Servers** is chosen, then the options defined in the HTTP Server window for predefined servers are ignored.

3. Define another rule as follows:

Source	Destination	Services	Action	Track	Install On
any	any	any	Reject	Long Log	Gateways

This rule prevents the use of HTTP without User Authentication on the gateway.

Restricting Incoming and Outgoing HTTP

Assume the following Rule Base:

Source	Destination	Services	Action	Track	Install On
AllUsers@any	localnet	http	UserAuth	Long Log	Gateways
AllUsers@localnet	any	http	UserAuth	Long Log	Gateways
any	any	any	Reject	Long Log	Gateways

The first rule requires users of incoming HTTP to authenticate before accessing internal HTTP servers defined in the HTTP servers window (FIGURE 1-14 on page 46).

The second rule applies to internal users accessing external HTTP. According to the second rule, internal users must be authenticated before accessing external HTTP.

The HTTP Security Server in Proxy Mode

For internal users, the behavior of the HTTP Security Server also depends on whether it is defined as the proxy to the users' Web browser (in the browser's proxy settings). There are two different proxy settings. Each setting has specific advantages and implications for local users.

- HTTP proxy

  The HTTP Security Server can be defined as the HTTP proxy to the user's Web browser. This provides several advantages regarding entering user passwords and managing authentication. For more information, see "The HTTP Security Server as an HTTP Proxy" on page 50.

- Security proxy

  The HTTP Security Server can be defined as the Security Proxy to the user's Web browser. The HTTP Security Server proxies HTTPS (HTTP encrypted by SSL) connections. Although it does not inspect content of HTTPS connections, the administrator can provide security by specifying User Authentication for outgoing HTTPS. For more information, see "HTTP Security Server as a Security Proxy — Authenticating Outgoing HTTPS" on page 51.

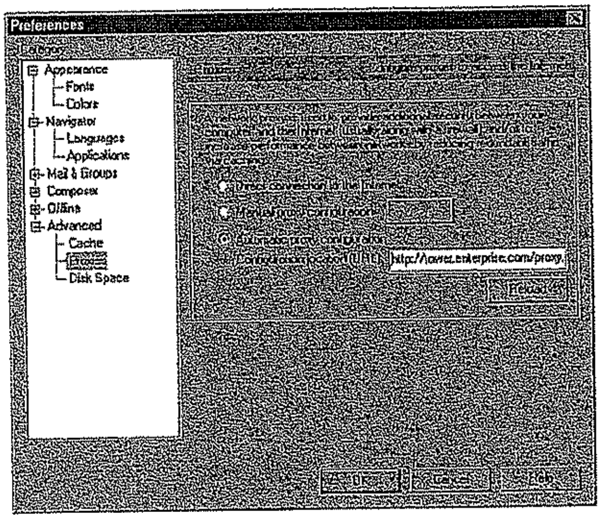
Note – Proxy settings are not mutually exclusive and can be used together.

The HTTP Security Server as an HTTP Proxy

Internal users can define the HTTP Security Server as the HTTP proxy to their Web browsers. The HTTP Security Server is defined as an HTTP proxy in the proxy settings of the user's Web browser.

Note – The FireWall-1 HTTP Security Server will only handle proxy requests for authenticated HTTP connections and resources. If you do not use authentication or resources, you cannot use the FireWall-1 HTTP Security Server as an HTTP proxy.

FIGURE 1-15 depicts the proxy settings window for Netscape 4.0.



○ Direct connection to the Internet
○ Manual proxy configuration
● Automatic proxy configuration
Configuration location (URL): http://tower.enterprise.com/proxy.

FIGURE 1-15 Defining the gateway as the HTTP proxy — Netscape 4.0

Defining the FireWall-1 HTTP Security Server as a proxy offers several advantages:

- You can centralize Authentication information to one location, the FireWall-1 machine.
- When defined as the HTTP proxy, the FireWall-1 HTTP Security Server can handle FTP requests through a Web browser. For more information, see "HTTP Security Server" on page 112.
- When a local user requests a URL in the Web browser, the browser sends the entire URL path to the HTTP Security Server.

  When the HTTP Security Server is not defined as a proxy, it only receives the name of the requested server from the Web browser. This means HTTP Security Server must retrieve the entire URL.

- All outgoing HTTP connections are mediated by the HTTP Security Server.

In addition, the FireWall-1 HTTP Security Server offers several advantages over other Authentication Servers.

- FireWall-1 supports S/Key, SecurID, RADIUS, TACACS and AXENT Defender authentication.
- FireWall-1 supports restriction by day of week and/or time of day.
- FireWall-1 can automatically expire a user according to the **Expiration Date** specified in the **General** tab of the **User Properties** window.

Note – Although it is defined as the HTTP proxy to the Web browser, the FireWall-1 HTTP Security Server is *not* an official HTTP proxy, that is, it does not relay HTTP traffic in the way that, for example, the CERN HTTP proxy does. It is not an independent daemon that runs outside of FireWall.

Caching — Using an HTTP Proxy behind the HTTP Security Server

A disadvantage to using the HTTP Security Server as a proxy is that it does not cache pages used by its client (the browser). If you wish to provide caching for HTTP users, you can put an HTTP proxy behind the FireWall-1 HTTP Security Server. To do this, you must specify the host and port of the proxy in the **HTTP Next Proxy** field in the **Security Servers** tab of the **Properties Setup** window.

HTTP Security Server as a Security Proxy — Authenticating Outgoing HTTPS

HTTPS (HTTP encrypted by SSL) connections can be handled by the HTTP Security Server when it is defined as the Security Proxy to the local user's Web browser. The HTTP Security Server proxies outgoing HTTPS connections, but does not inspect content. This option is used to authenticate internal users accessing external HTTPS.

The user can configure a Security Proxy for the following Web browsers:

- Internet Explorer version 3.0x and higher
- Netscape version 4.0x and higher

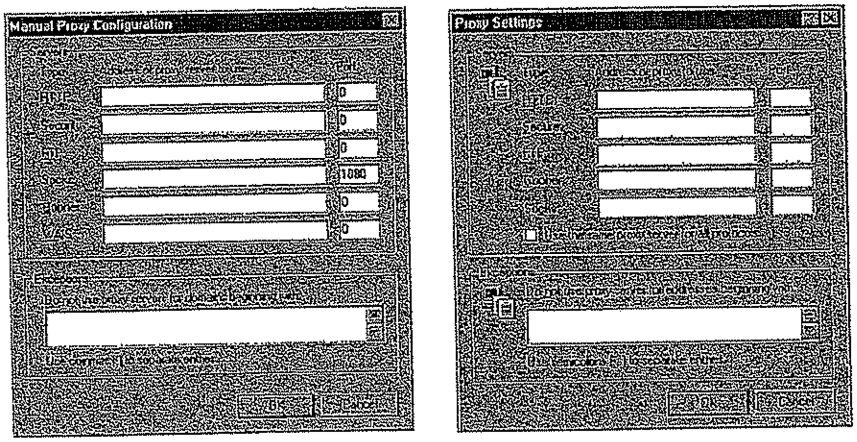


FIGURE 1-16 HTTP Proxy and Security Proxy Settings — Netscape 4.0x and Internet Explorer 3.0x

HTTPS requests generally use the HTTP "CONNECT" method (tunneling mode). Because the CONNECT method only specifies a hostname and port, the HTTP Security Server does not have access to the content of the communication, not even the URL. In addition, the Security Server does not verify that the connections are really using HTTPS — it only checks the requested hostname and port number. All communication between the client and the target server is encrypted — the HTTP Security Server only proxies the connection. This is useful if internal users want to send encrypted information over the Internet.

In Security Proxy mode, you can provide security by requiring internal users to authenticate before accessing external HTTPS servers.

Note – Although the connection is encrypted between the local client and the external server, the authentication session between the local client and the HTTP Security Server is clear (unencrypted).

Tip – If you are using the HTTP Security Server as a Security Proxy, you can also provide security for HTTPS connections using Resource rules. For more information, see Chapter 2, "Security Servers" of this book.

Authenticating Internal Users Accessing External HTTPS (Security Proxy Mode)

To authenticate users of outgoing HTTPS, proceed as follows:

1 Internal users must define the HTTP Security Server as the Security Proxy on port 443. This is done in the proxy settings of the user's Web browser (FIGURE 1-16 on page 52).

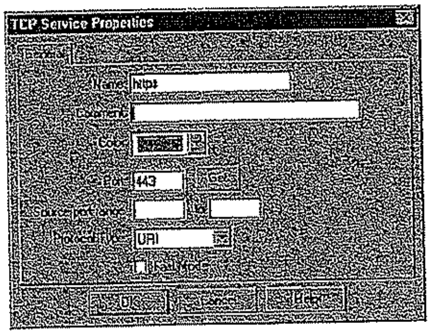
2 Add the following line to the file `$FWDIR/conf/fwauthd.conf`:

```
443   bin/in.ahttpd   wait   0
```

This enables the HTTP Security Server to run on the port specified for the Security Proxy.

3 In the HTTPS service properties, set the **Protocol Type** to **URI** (FIGURE 1-17).

This assures that the HTTPS service (using port 443) will be mediated by the HTTP Security Server.



Name: https

Comment:

Port: 443

Source port range: to

Protocol type: URI

□ Fail Mode

FIGURE 1-17 HTTPS Service Definition

4 Define a rule similar to the following

Source	Destination	Services	Action	Track	Install On
All_Users@localnet	any	https	UserAuth	Long Log	Gateways

According to this rule, internal users must authenticate before accessing external HTTPS.

5 In the rule's **User Authentication Action Properties** window, check **All Servers** under **HTTP**.

This assures that the outgoing connections will be allowed to continue from the HTTP Security Server to any external host or port.

Tip – You can also configure the HTTP Security Server to encrypt and decrypt HTTPS connections on the gateway. For more information, see "Support For HTTPS — Controlling External Access to Internal HTTPS" on page 60.

Configuring Multiple HTTP Security Servers

You can modify the Security Server configuration to enable multiple HTTP Security Servers to run concurrently. Multiple HTTP clients connecting concurrently can be handled by several HTTP Security Servers.

Note – This option is recommended for gateway machines with more than one CPU. Using this option on a gateway machine with only one CPU will result in a performance degradation.

The file `$FWDIR/conf/fwauthd.conf` lists the Security Server executables and their assigned port numbers. The example line below shows the HTTP Security Server assigned to a dynamically allocated port:

```
80    bin/in.ahttpd     wait    0
```

The last field indicates the Security Server port number.

A negative port value indicates that FireWall-1 randomly chooses multiple ports for the HTTP Security Server. The absolute value indicates the number of random ports that will be chosen. The number of random ports should correspond to the number of CPUs the gateway machine has.

The example below indicates that FireWall-1 will randomly select four high ports for the HTTP Security Server:

```
80    bin/in.ahttpd     wait    -4
```

According to the above example, an HTTP client will initially connect to one of four randomly selected ports. If the same client connects again before the **Authorization Timeout** specified in the **Security Servers** tab of the **Properties Setup** window, the same port will be chosen. If the same client connects again after the **Authorization Timeout**, another port will be chosen.

Another concurrent HTTP client will connect to one of the remaining free ports.

For more information, see "fwauthd.conf file" on page 123.

HTTP Security Server — When the User Connects

Password Prompt

When a user is intercepted by the HTTP Security Server, a password prompt window is displayed in which the user is asked to enter a user id and a password. The format of the window depends on the HTTP browser in use, since it is the browser that displays the window, not FireWall-1. However, some of the data displayed in the window is supplied to the browser by the FireWall-1 HTTP Security Server.

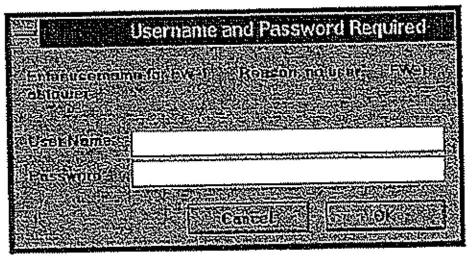


FIGURE 1-18 A Typical User ID and Password Window

The information given in the password prompt window usually includes:

- the Authentication scheme required by FireWall-1
- whether Authentication is required for the HTTP server, and if so, the server's realm name
- a "reason" message giving the reason for the last Authentication failure

Multiple Users and Passwords

This applies only to when the HTTP Security Server is not being used as a user's Web proxy. This is relevant for external users accessing internal HTTP, and internal users who have not defined the HTTP Security Server as a proxy.

In HTTP, the Web browser automatically supplies the user's password to the server once the user authenticates. If the user requests another server, the browser cannot send the password to the new server, and the user must reauthenticate.

The user can specify different user names (and passwords) at the password prompt for the HTTP server and FireWall-1, as follows:

```
server_username@FireWall-1_username
```

In the same way, the user can enter two passwords, as follows:

```
server_password@FireWall-1_password
```

If there is no password for the server, only the FireWall-1 password should be entered.

If the user enters one user name and two passwords, the same user name is used for both the HTTP server and FireWall-1, but the different passwords are used as indicated.

If @ is part of the password, the user should type it twice (for example if the password is mary@home, type mary@@home).

"Reason" Messages

Authentication attempts may be denied for any of the following reasons. The browser displays these messages in the password prompt.

TABLE 1-4 HTTP Security Server "Reason" messages

error	meaning
no user	No user id was entered.
no password	No password was entered.
wrong password	The OS or FireWall-1 password was incorrect.
S/Key	S/Key authentication failed.
SecurID	SecurID authentication failed.
WWW server	The FireWall-1 password was correct, but the server did not authorize the user (probably because the server password was incorrect).
user limitations	The user is not authorized for the given day of week, time of day, source or destination, or the user has expired.
FW-1 rule	The FireWall-1 password was correct, but the user was not authorized because there was no matching rule in the Rule Base.

In addition to "Reason" messages, additional messages may be displayed by the browser.

TABLE 1-5 Browser Messages

message	meaning and/or corrective action
Failed to connect to the WWW server	Notify your system manager.
Unknown WWW server	FireWall-1 could not determine to which server the URL should be sent. This can happen for two reasons: The URL is incorrect, or there is a resolution problem.
Authentication Services are unavailable	Notify your system manager.
FireWall-1 is currently busy - try again later	Wait and try again. If the problem persists, notify your system manager.
Simple requests (HTTP 0.9) are not supported	HTTP 0.9 clients are not supported by the FireWall-1 HTTP Security Server. Note that HTTP 0.9 servers are supported.

Additional messages may be displayed if FireWall-1 encounters an error, such as no OS password defined for a user who is required to supply a OS password, or problems with SecurID server etc. In the event one of these messages is encountered, the user should notify the system administrator.

Differences Between Clients

Some clients do not display authorization information in the password prompt window, or display only some of the information.

When using such a client, the user can sometimes view the missing information by clicking on **Cancel** (to view the information) and then **Reload** (to display the password prompt window again), or by performing the equivalent functions.

If the password prompt window does not display what kind of password must be entered (this can happen if the user name is unknown), the user should enter the user name without the password and click on **OK**. If the client requires a non-empty password, the user should type a single space. The window will be displayed again, including the password type.

HTTP Security Server — Security Considerations

In HTTP, the Web browser automatically supplies the password to the server for each connection. This creates special security considerations when using the HTTP Security Server with one-time passwords.

To avoid forcing users of one-time passwords to generate a new password for each connection, the HTTP Security Server extends the validity of the password for the time period defined in **Authorization Timeout** in the **Authentication** tab of the **Properties Setup** window. Users of one-time passwords do not have to reauthenticate for each request during this time period.

Each successful access resets the timer to zero. Because the authorization period is renewable, and the Web browser keeps supplying the password, the time period during which a one-time password can be used can be unlimited.

This problem can be solved by using the Reauthentication options in the HTTP Server definition ("HTTP Servers List (Security Servers tab)" on page 46). For example, you can specify that every request to a specific HTTP server requires a new password, or that requests that change a server's configuration require a new password. For more information, see "HTTP Servers List (Security Servers tab)" on page 46.

HTTP Security Server and Non-Transparent Authentication

This section describes how the HTTP Security Server is configured when implementing Non-transparent Authentication. In Non-transparent Authentication, the user must explicitly connect to the FireWalled gateway in order to authenticate before continuing to the target host.

Non-transparent Authentication is implemented when the `prompt_for_destination` parameter is set to `true` in the `objects.C` file. This value indicates that if the user requests the gateway as the destination, the FireWall Module assumes the user's "true" destination is some other server, and prompts the user for the "true" destination. For information on Non-transparent authentication, see "Non-Transparent User Authentication" on page 42.

Tip – The HTTP Security Server also supports HTTPS in non-transparent mode. For more information, see "Support For HTTPS — Controlling External Access to Internal HTTPS" on page 60

Configuring URLs

Suppose that in the configuration depicted in FIGURE 1-19 on page 59, there are HTTP servers on all the hosts (Tower, Palace, and BigBen) which are protected by a FireWall-1 HTTP Security Server on the gateway London.

The URLs should be set up as follows:

http://*gateway*/*logical server name*/...

where the ellipsis (...) indicates the part of the URL that the server receives and parses. This is usually a file name.

For example, assume HTTP servers running on London-net, as listed in FIGURE 1-19.

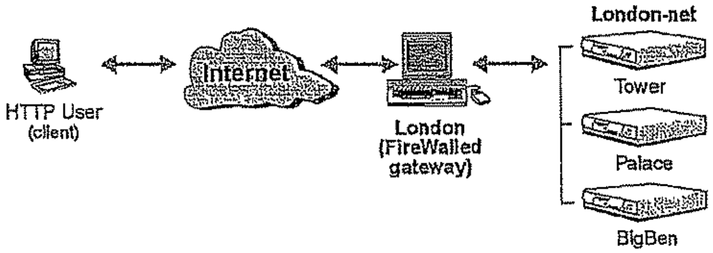


FIGURE 1-19 HTTP Servers behind a FireWalled gateway

TABLE 1-6 Servers and URLs

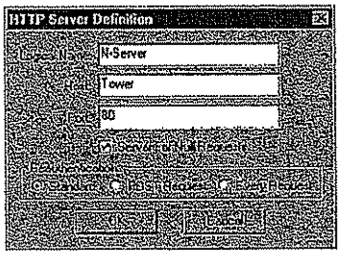
server name	URL	host	port
info	http://www.London.com/info/info.html	BigBen	80
tickets	http://www.London.com/tickets/ordrtick.html	Palace	80
actors	http://www.London.com/actors/bios.html	Tower	8000
reviews	http://www.London.com/reviews/clips.html	Tower	8080

In this case, the gateway is `www.London.com` and the server names are those specified under **HTTP Servers** in the **Security Servers** tab of the **Properties Setup** window (see FIGURE 1-13 on page 45), where the connection between the server name and the host and port is defined. The only externally known name is `www.London.com`.

For example, a user who wishes to access the HTTP server "actors" on the host Tower, must specify a URL of `http://www.London.com/actors/bios.html` (where `bios.html` is a specific page maintained by the server). The connection is intercepted by the HTTP Security Server on the gateway London.

Configuring a Server for Null Requests

A Server for Null Requests is used when the URL is of the form `http://`*`gateway`*`/` or `http://`*`gateway`*. In this case, the URL passed to the server is `/`. In Non-transparent Authentication, this allows a user to connect to the gateway without having to specify the name of a target server behind the gateway.. The Server for Null Requests is defined using the **HTTP Server Definition** window (FIGURE 1-20).



HTTP Server Definition

Logical Name: N-Server

Host: Tower

Port: 80

☑ Server for Null Requests

Reauthentication:

● Standard ○ POST Requests ○ Every Request

FIGURE 1-20 HTTP Server Definition — Server for Null Requests

For example, a user connects to London, the FireWalled gateway, as follows:

```
http://www.london.com
```

If you have specified an HTTP server as a **Server for Null Requests** and checked **Predefined Servers** in the **User Authentication Action Properties** window of the relevant rule, the HTTP Security Server on the gateway directs the connection to the Server for Null Requests.

You can configure this server to display a Web page with links to internal servers. This server then provides an "entry point" to other internal servers. In this way, the user does not have to know the names of the target servers behind the gateway. All the user needs to know is the name of the FireWalled gateway.

Support For HTTPS — Controlling External Access to Internal HTTPS

If you are implementing Non-transparent Authentication, the HTTP Security Server can be configured to encrypt and decrypt HTTPS connections. This option enables the HTTP Security Server to inspect the contents of HTTPS connections.

The connection can be encrypted between the client and the HTTP Security Server, and then possibly again from the HTTP Security Server to the target server. For example, you can specify that connections between the client and HTTP Security Server are encrypted. The HTTP Security Server on the gateway decrypts and inspects the connection. The connection can then be encrypted again from the HTTP Security Server to the target host. The authentication session is encrypted as well.

This option is known as "Non-transparent Mode" because the user of HTTPS must request the gateway before being allowed to continue to the target host. Because the HTTP Security Server is not defined as a Security Proxy to the user's Web browser, Non-transparent Mode is best used to authenticate external users accessing internal servers.

To configure support for HTTPS, proceed as follows:

Generating CA Keys

You must first generate the CA Key pair to be used by the FireWall-1 Management Station and the gateway:

1 Generate the CA Key for the Management Station using the `fw ca genkey` command as follows:

```
fw ca genkey "[-ou] [-o] [-c]"
```

where `-ou`, `-o`, and `-c` specify the Distinguished Name (DN) of the Certificate Authority.

2 Distribute the CA Key to the FireWalled gateway using the `fw ca putkey` command.

```
fw ca putkey <target> [-p password]
```

The parameter `target` is the IP address or resolvable name of the machine on which you are installing the CA key (the FireWalled gateway).

The parameter `-p password` is a password that will be used to authenticate future communication between the Management Station and the gateway.

If you do not enter a password, you will be prompted for one.

3 Generate a Certificate using the following command:

```
fw certify ssl <management> <target> [-p password]
```

The parameter `management` is the IP address or resolvable name of the gateway's Management station.

You must enter the same password you used when you issued the `fw ca putkey` command in step 2.

Modifying the Security Server Configuration File

4 You must next modify the file `$FWDIR/conf/fwauthd.conf` by adding a line which enables the HTTP Security Server to run on an additional service port dedicated to HTTPS. According to the example below, HTTPS connections will connect to the gateway on port 443.

Example:

```
443    bin/in.ahttpd    wait    0   ec
```

The last field specifies what to do with HTTPS (SSL) connections. TABLE 1-7 lists the available options:

TABLE 1-7 HTTPS options

option	meaning
ec	encrypt connections between the client and the gateway
es	encrypt connections between the gateway and the server
eb	encrypt both — encrypt connections between the client and the gateway, and then between the gateway and the server.
ns	no SSL (no encryption)

Leaving this field empty is the same as specifying `ns`.

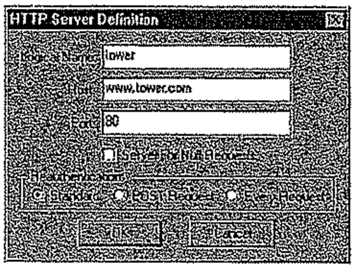
For more information about the file `$FWDIR/conf/fwauthd.conf`, see "Security Server Configuration" on page 123.

HTTP Servers

5 You must also define the HTTP servers to which HTTPS connections are allowed.

HTTP servers are defined in the **HTTP Server Definition** window (FIGURE 1-21). For information on defining HTTP servers, see "HTTP Servers List (Security Servers tab)" on page 46 in this chapter.

Example



HTTP Server Definition

Logical Name: tower

Host: www.tower.com

Port: 80

□ Server for Null Requests

Reauthentication

● Standard ○ POST Request ○ Every Request

FIGURE 1-21 Example HTTP Server definition

You must specify the **Logical Name, Host** and **Port** number on which you installed the servers which will handle HTTPS connections.

HTTPS Service Properties

6 You must next modify HTTPS properties to assure that the service will be mediated by the HTTP Security Server. In the HTTPS service definition, set the **Protocol Type** to **URI** (see FIGURE 1-17 on page 53).

User Authentication Rule

7 Next, define a rule similar to the following:

Source	Destination	Services	Action	Track	Install On
All_Users@any	localnet	https	UserAuth	Long Log	Gateways

In the rule's **User Authentication Action Properties** window, specify **Predefined Servers** under **HTTP**. This restricts incoming HTTPS to the servers listed in the **Security Servers** tab of the **Properties Setup** window (the HTTP Servers you defined in step 5 on page 62).

How the User Connects

An external user of HTTP must specify the name of the FireWalled gateway and the logical name of the target server in the requested URL. This assures that the request will be intercepted by the HTTP Security Server on the gateway. The URL is set up as follows:

```
https://<gateway_name>/<logical_server_name>/...
```

For example, if the gateway name is London, and the target server (behind London) is Tower, then the user specifies the following URL:

```
https://www.london.com/tower/...
```

For information on how to set up URLs for Non-transparent Authentication, see "Configuring URLs" on page 58.

Putting Existing HTTP Servers Behind the HTTP Security Server

This section applies to users who wish to implement Non-transparent authentication.

To put an existing HTTP server behind the FireWall-1 HTTP Security Server, proceed as follows:

▼ If you have only one HTTP server, and it is on your gateway

1. Replace the HTTP server with the FireWall-1 HTTP Security Server.
2. Put the HTTP server elsewhere.
3. For security reasons, it is recommended that you put the HTTP server on a different computer. However, you can also put it on a different port on the same computer.
4. Update the **HTTP Servers** list in the **Security Servers** tab of the **Properties Setup** window in accordance with where you put the HTTP server (see previous step).

▼ If you have only one HTTP server, and it is *NOT* on your gateway

1. Arrange for the server address to be directed to the FireWall-1 HTTP Security Server (host name) on the gateway.

   This is done outside of FireWall-1, either by publicizing the new address for the existing host name, or by creating a new host name and notifying your authorized users.

2. Add the server to the list of HTTP servers in the **Control Properties/Security Servers** window.

   Even if there is only one server behind FireWall-1, you should still give it a name and add it to the HTTP server table in the **Control Properties/Security Servers** window (see "HTTP Servers List (Security Servers tab)" on page 46). However, the server name may be omitted from the URLs that refer to it (if there is only one server), so there is no need to change URLs when putting a single server behind FireWall-1 if the host name for the FireWall and the server are the same.

▼ If you have more than one HTTP server

1. Arrange for the server address to be directed to the FireWall-1 HTTP Security Server (host name) on the gateway.
2. This is done outside of FireWall-1, either by publicizing the new addresses for the existing host name, or by creating a new host name and notifying your authorized users.
3. Add the servers to the list of HTTP servers in the **Control Properties/ Security Servers** window.
4. Modify all the HTML source code so that absolute references point to the appropriate servers.

   This step is necessary for non-transparent authentication only.

   The first field of an absolute URL should be replaced by "*gateway/logical server name/*".

   It is not necessary to modify relative references.

Session Authentication

Overview

Session Authentication can be used to authenticate users of any service. FIGURE 1-22 shows what happens when a rule's Action specifies Session Authentication.

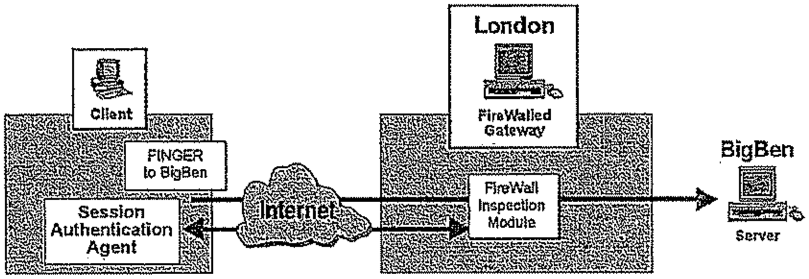


FIGURE 1-22 Session Authentication

The Session Authentication process is as follows:

1. The user initiates a connection directly to the server.
2. The FireWall-1 Inspection Module intercepts the connection. The Inspection Module connects to a Session Authentication Agent on the client.

   In the above configuration, the Session Authentication Agent is running on the client, but it can run on another machine (on any of the supported platforms).
3. The Session Authentication Agent prompts the user for authentication data and returns this information to the Inspection Module.
4. If the authentication is successful, then the FireWall module allows the connection to pass through the gateway and continue on to the target server.

In contrast to User Authentication, Session Authentication does not result in an additional connection to the server. The advantage of Session Authentication is that it can be used for every service. It requires a Session Authentication Agent which prompts the user through a series of pop-up screens.

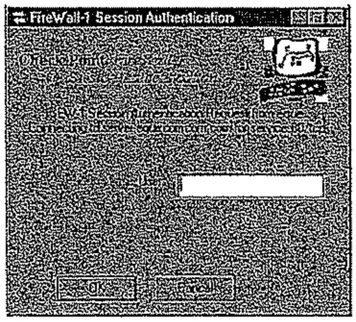


FIGURE 1-23 FireWall-1 Session Authentication Agent Prompt

The Session Authentication Agent is an application that communicates with the FireWall Module using the FireWall-1 Session Authentication Agent Protocol. The Session Authentication Agent can be running on the following network objects:

- the source of the connection (i.e., the client that initiated the connection)
- the destination of the connection
- a specific host

Session Authentication — Deployment

Example Configuration

In the configuration depicted in (FIGURE 1-24), all localnet users must be authenticated before accessing the Internet.

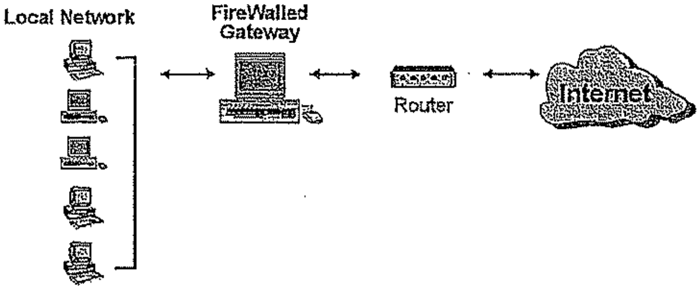
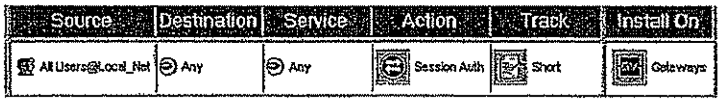


FIGURE 1-24 Example configuration

The following rule allows users of any service external access after successful Session Authentication.



Source	Destination	Service	Action	Track	Install On
All Users@Local_Net	Any	Any	Session Auth	Short	Gateways

FIGURE 1-25 Example Session Authentication Rule

Configuring Session Authentication

To enable Session Authentication for this configuration, the Administrator must do the following:

- install and configure the Session Authentication Agent
- define user properties
- define Session Authentication rule properties
- define logging and tracking

Configuring the Session Authentication Agent

Windows

To install the Session Authentication agent for Windows, run the `SETUP` program in the `DESKTOP PRODUCTS\SESSIONAGENT` directory on the CD-ROM.

OpenLook

A sample Session Authentication Agent for the OpenLook GUI is in `$FWDIR/bin/fwsngui`.

Opening the Session Authentication Agent

To open the Session Authentication agent, double-click on its icon in the system tray. The **FireWall-1 Session Authentication** window (FIGURE 1-26) is displayed.

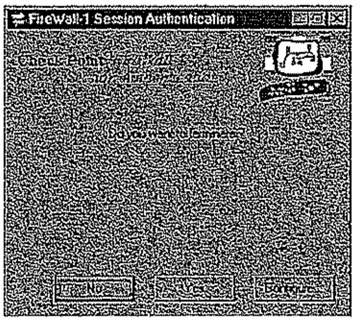


FIGURE 1-26 FireWall-1 Session Authentication window

Perform one of the following:

- To terminate the Session Authentication agent, click on **Yes**.
- To configure the Session Authentication agent, click on **Configure**.
- To close the FireWall-1 **Session Authentication** window, click on **No**.

Configuration

When you click on **Configure** in the **Session Authentication** window, the **Configuration** window (FIGURE 1-27) is displayed.

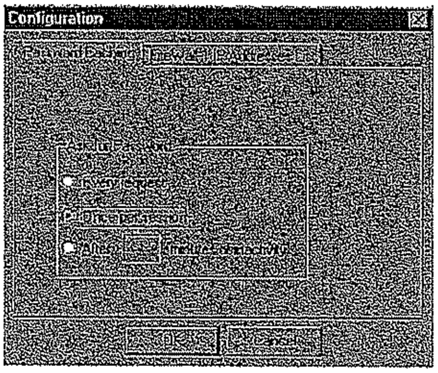


FIGURE 1-27 Configuration window — Password Caching tab

The **Configuration** window has two tabs, explained below.

Password Caching Tab

The **Password Caching** tab of the **Configuration** window enables you to specify how frequently the user is asked to supply a password (that is, to authenticate himself or herself). One-time passwords (such as SecurID) cannot be cached.

Check one of the available choices:

Every request — The user will be prompted for the password each time the FireWall Module requests authentication.

Each time the user initiates a session to which a Session Authentication rule applies, the user will be prompted for a password. In this case, no password caching occurs.

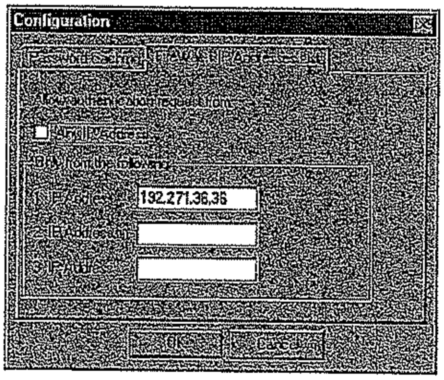
Once per session — The user will be prompted for a password once per Session Authentication agent session.

In this case, the user supplies the password once and the Session Authentication agent caches the password indefinitely. This option cannot be used with one-time passwords.

If the Session Authentication agent is terminated and then re-started, the user will have to supply the password again.

After ... minutes of inactivity — This option is the same as **Once per session**, except that the user will be prompted again for a password if there has been no authentication request for the specified time interval.

FireWall-1 IP Addresses List Tab



192.271.36.36

FIGURE 1-28 Configuration window — FireWall-1 IP Addresses List tab

The **FireWall-1 IP Addresses List** tab of the **Configuration** window enables you to specify the FireWall Modules for which this Session Authentication agent may provide authentication services.

Any IP Address — This Session Authentication agent may provide authentication services for any FireWall Module.

IP Address — This Session Authentication agent may provide authentication services only for a FireWall Module running on the specified IP address.

You can specify up to three IP addresses.

Pre-configuration

The `SETUP.INI` file in the `DESKTOP PRODUCTS\SESSIONAGENT` directory enables you to pre-configure the Session Authentication agent. This feature is useful if you plan to distribute the Session Authentication agent to many users and you do not want them to configure the agent themselves.